

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SOFTWARE

Tema:

“Implementación de la Fase 2 del servicio ‘Portafolio Estudiante’ en la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo SCRUM y validando su calidad en uso conforme a la norma ISO/IEC 25022.”



AUTOR: Jordan Steven Puruncajas Castillo

DIRECTOR: Ing. José Antonio Quiña Mera, PhD.

Ibarra-Ecuador

2026

UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA

IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1050319951		
APELLIDOS Y NOMBRES:	PURUNCAJAS CASTILLO JORDAN STEVEN		
DIRECCIÓN:	IBARRA, IMBABURA		
EMAIL:	jspuruncajasc@utn.edu.ec / castillo10steven@hotmail.com		
TELÉFONO FIJO:		TELF. MOVIL	0962683340

DATOS DE LA OBRA	
TÍTULO:	Implementación de la fase 2 del servicio 'Portafolio Estudiante' en la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo SCRUM y validando su calidad en uso conforme a la norma ISO/IEC 25022.
AUTOR:	Puruncajas Castillo Jordan Steven
FECHA DE APROBACIÓN:	31/03/2026
CARRERA/PROGRAMA:	<input checked="" type="checkbox"/> GRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN SOFTWARE
DIRECTOR:	Ing. Quiña Mera José Antonio, PhD.

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 31 días, del mes de marzo de 2026

EL AUTOR:

.....
Jordan Steven Puruncajas Castillo
1050319951

CERTIFICACIÓN DEL DIRECTOR

Ibarra, 31 de marzo de 2026

Ing. José Antonio Quiña Mera, Ph.D.
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

.....
Ing. José Antonio Quiña Mera, Ph.D.
Director de Trabajo de Grado
C.C.: 1002322384

DEDICATORIA

A las personas, quienes con amor, paciencia y apoyo han caminado junto a mí en cada momento de este viaje.

A mi mamá, por ser el centro que impulsa mis decisiones, por su amor incondicional, su sabiduría y por enseñarme que cada dificultad es una oportunidad para avanzar y que la constancia lleva a alcanzar los sueños.

A mi hermano y mis hermanas, por estar en los momentos difíciles, por su apoyo y por hacerme saber que nunca estoy solo en este trayecto.

A mis amigos, los que el tiempo demostró que son hermanos de verdad y los que conocí en estos años, por los momentos divertidos, el apoyo real y por hacer que cada etapa de este camino sea un recuerdo que guardaré siempre.

A todos ustedes, que son parte de este logro, les dedico con mucho agradecimiento y cariño todo lo que he conseguido.

AGRADECIMIENTO

A mi mamá, Ana, por su amor incondicional, por ser mi mayor apoyo y por enseñarme que con esfuerzo y dedicación se puede lograr lo que nos proponemos. Su fortaleza y ejemplo me han motivado a seguir adelante en cada momento difícil.

A mi tutor de tesis, Antonio, y a mi asesora, Cathy, por guiarme, confiar en mí y darme la oportunidad de demostrar lo que soy capaz en un trabajo tan importante.

A todos mis profesores, por compartir su conocimiento conmigo, por mostrarme la importancia de seguir aprendiendo y por motivarme a crecer tanto en lo profesional como en lo personal. Cada clase y cada conversación dejó algo importante en mi formación.

A mis compañeros, por estar ahí en los buenos y malos momentos, por las tardes de estudio juntos y por hacer que esta experiencia universitaria fuera más que solo clases. Sus ánimos fueron un impulso cuando más lo necesitaba.

A cada uno de ustedes, mi más sincero agradecimiento, pues sin su apoyo este logro no habría sido posible

RESUMEN

La Universidad Técnica del Norte dispone de la aplicación móvil institucional UTN-Móvil, la cual integra distintos servicios académicos para la comunidad universitaria. El servicio "Portafolio Estudiante" requería la incorporación de funcionalidades adicionales que complementaran las desarrolladas en su primera fase, con el propósito de ampliar la cobertura académica y mejorar la experiencia del usuario en dispositivos móviles. El presente trabajo tuvo como objetivo implementar la Fase 2 del servicio "Portafolio Estudiante" dentro de la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo ágil SCRUM y evaluando la calidad en uso conforme a la norma ISO/IEC 25022. La investigación fue de carácter aplicado. Para el desarrollo se adoptó SCRUM, estructurado en seis sprints, utilizando Flutter con una arquitectura limpia como framework frontend multiplataforma y con microservicios como base del backend, integrados a la base de datos Oracle institucional. Se implementaron ocho nuevas funcionalidades: foro en línea, evaluación en línea, contenidos de la materia, formatos vinculación, evaluación docente, avisos modales, pagos y becas. La calidad en uso fue medida mediante un taller práctico y la encuesta SUS, obteniendo un resultado global de 80.98%, puntaje que se ubica dentro del rango satisfactorio según la norma ISO/IEC 25022. Se concluye que la Fase 2 fue implementada exitosamente, demostrando que las nuevas funcionalidades son intuitivas, funcionales y aceptadas positivamente por los estudiantes, consolidando a la aplicación móvil como una herramienta de calidad para la gestión académica universitaria.

Palabras clave: aplicación móvil, Flutter, microservicios, SCRUM, ISO/IEC 25022, calidad en uso, gestión académica.

ABSTRACT

The Universidad Técnica del Norte provides the UTN-Móvil institutional mobile application, which offers a range of academic services to the university community. The "Student Portfolio" service required the addition of complementary functionalities to those developed in its first phase, aiming to extend its academic coverage and improve the user experience on mobile devices. The objective of this work was to implement Phase 2 of the "Student Portfolio" service within the UTN mobile platform, applying the SCRUM agile framework and evaluating quality in use in accordance with the ISO/IEC 25022 standard. The research was applied in nature. Development followed the SCRUM methodology, organized into six thematic sprints, using Flutter with a clean architecture as a cross-platform frontend framework and microservices as the backend foundation, all integrated with the institutional Oracle database. Eight new functionalities were implemented: online forum, online assessment, course content, community outreach formats, teacher evaluation, modal notices, payments, and scholarships. Quality in use was assessed through a practical workshop and the SUS questionnaire, yielding an overall result of 80.98%, a score within the satisfactory range per-ISO/IEC 25022. It is concluded that Phase 2 was successfully implemented, demonstrating that the new functionalities are intuitive, functional, and positively accepted by students, consolidating the mobile application as a quality tool for university academic management.

Keywords: Mobile Application, Flutter, Microservices, SCRUM, ISO/IEC 25022, Quality in use, Academic Management.

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
Problema de investigación	1
Objetivos	2
Objetivo general	2
Objetivos específicos	2
Alcance	2
Metodología	3
Justificación	4
Contexto de Investigación.....	5
I. MARCO TEÓRICO	1
1.1 Sistema informático integrado universitario (SIIU)	1
1.1.1 El "portafolio estudiante" en el contexto del SIIU de la UTN	1
1.1.2 Avance del Servicio "Portafolio Estudiante": De la Fase 1 a los Nuevos Requisitos	1
1.1.3 Pertinencia del desarrollo y su alineación con los objetivos institucionales	2
1.2 Arquitectura de microservicios.....	2
1.2.1 Principios de diseño, ventajas y desafíos de los microservicios	2
1.2.2 Desarrollo backend con NestJS y Typescript para microservicios	3
1.2.3 Interfaces de programación de aplicaciones (APIs REST) y Comunicación	3
1.2.4 Integración de microservicios con bases de datos (Oracle)	3
1.3 Aplicaciones móviles	4
1.3.1 Desarrollo multiplataforma con Flutter y lenguaje Dart	4
1.3.2 Arquitectura limpia en aplicaciones Flutter	5
1.3.3 Desarrollo iterativo y evolución funcional de aplicaciones móviles	6
1.3.4 Consideraciones de seguridad y privacidad en aplicaciones móviles	7
1.3.5 Estrategias para la gestión de errores y resiliencia en aplicaciones móviles	8
1.4 Metodología de desarrollo de software	9
1.4.1 Fundamentos del Desarrollo Ágil y el Manifiesto Ágil	9
1.4.2 El Marco de Trabajo Scrum y su Proceso de Desarrollo	10
1.4.3 Aplicación de Scrum en proyectos de desarrollo de software	11
1.5 Calidad del software y modelos de evaluación.....	12
1.5.1 La Familia de normas ISO/IEC 25000 (SQuaRE)	12
1.5.2 Modelo de Calidad de la Norma ISO/IEC 25010	13
1.5.3 Calidad en uso	14
1.5.4 Norma ISO/IEC 25022: Medición de la Calidad en Uso	14
1.5.5 Escala de Medición de la Calidad en Uso.....	14
1.6 Trabajos relacionados.....	15
II. DESARROLLO	19
2.1 DESARROLLO FASE - PRE-JUEGO	19
2.1.1 Asignación de Roles del Equipo Scrum	19

2.1.2	Definición de Requisitos	20
2.2	Desarrollo Fase de Juego.....	25
	Sprint 1: Pagos y Contenidos de la materia	25
	Sprint 2: Formatos Vinculación y Avisos Modales	30
	Sprint 3: Foro Online	36
	Sprint 4: Evaluación Docentes.....	43
	Sprint 5: Becas.....	48
	Sprint 6: Evaluación Online.....	56
2.3	Desarrollo Fase de Post-Juego.....	63
III.	VALIDACIÓN DE RESULTADOS	64
3.1	Definición de Modelo de Calidad en Uso	64
3.2	Medición de Calidad en Uso	64
3.2.1	Definición de Muestra poblacional.....	65
3.2.2	Taller Práctico	65
3.2.3	Encuesta SUS	66
3.3	Validaciones estadísticas de los instrumentos de medición.....	68
3.3.1	Fiabilidad del taller práctico	68
3.3.2	Fiabilidad de la encuesta SUS	68
3.4	Evaluación del Modelo de Calidad en Uso	69
3.4.1	Eficacia.....	69
3.4.2	Eficiencia.....	70
3.4.3	Satisfacción.....	71
3.5	Resultados del Modelo de Calidad en Uso.....	73
	CONCLUSIONES.....	75
	RECOMENDACIONES	76
	REFERENCIAS	77
	ANEXOS.....	83
Anexo A.	Fotografías de aplicación del taller práctico y encuesta SUS	83
Anexo B.	Taller práctico para evaluar la aplicación móvil	85
Anexo C.	Tabulación del taller práctico.....	86
Anexo D.	Tabulación encuesta SUS	89

ÍNDICE DE TABLAS

Tabla 1	Trabajos Relacionados	5
Tabla 2	Asignación de Roles	19
Tabla 3	Historia de Usuario 1: Pagos	20
Tabla 4	Historia de Usuario 2: Contenidos de la Materia.....	20
Tabla 5	Historia de Usuario 3: Formatos de Vinculación	21
Tabla 6	Historia de Usuario 4: Avisos Modales.....	21
Tabla 7	Historia de Usuario 5: Foro General	22
Tabla 8	Historia de Usuario 6: Foro Actividad	22
Tabla 9	Historia de Usuario 7: Evaluación Docentes.....	22
Tabla 10	Historia de Usuario 8: Becas	23
Tabla 11	Historia de Usuario 9: Evaluación Online	23
Tabla 12	Product Backlog	24
Tabla 13	Sprint Backlog – Sprint 1	25
Tabla 14	Criterios de Aceptación Sprint 1.....	29
Tabla 15	Sprint Backlog – Sprint 2	30
Tabla 16	Criterios de Aceptación Sprint 2.....	35
Tabla 17	Sprint Backlog – Sprint 3	36
Tabla 18	Criterios de Aceptación Sprint 3.....	42
Tabla 19	Sprint Backlog – Sprint 4	43
Tabla 20	Criterios de Aceptación Sprint 4.....	48
Tabla 21	Sprint Backlog – Sprint 5	48
Tabla 22	Criterios de Aceptación Sprint 5.....	55
Tabla 23	Sprint Backlog – Sprint 6	56
Tabla 24	Criterios de Aceptación	62
Tabla 25	Modelo de calidad en uso	64
Tabla 26	Taller práctico.....	65
Tabla 27	Preguntas Encuestas SUS	67
Tabla 28	Peso respuesta encuesta SUS.....	71
Tabla 29	Resultado SUS - Utilidad	71
Tabla 30	Resultados SUS - Comodidad	72
Tabla 31	Resultados Finales de Calidad en Uso - Fase 2	74

ÍNDICE DE FIGURAS

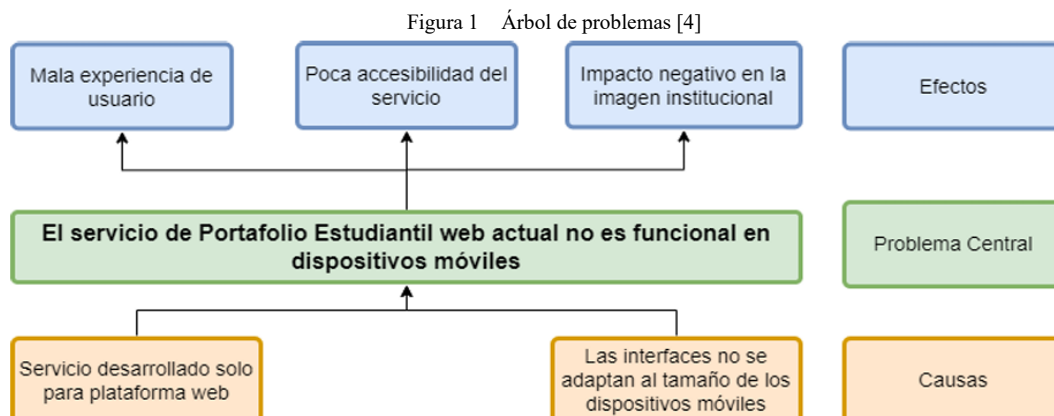
Figura 1	Árbol de problemas [4].....	1
Figura 2	Arquitectura propuesta [4].....	3
Figura 3	Diagrama de Arquitectura Limpia [39].	5
Figura 4	Fases del Proceso de Desarrollo [57].....	10
Figura 5	Divisiones de la familia de normas ISO/IEC 25000 [58].	13
Figura 6	Escala de medición de Calidad en Uso [62].	15
Figura 7	Diagrama de las Fases de Desarrollo.....	19
Figura 8	Listado del historial de transacciones del estudiante	26
Figura 9	Detalle expandido de una transacción específica.....	27
Figura 10	Scroll infinito para la lista de transacciones disponibles	28
Figura 11	Árbol de contenidos de la materia	28
Figura 12	Vista de la planificación curricular.....	29
Figura 13	Listado de los formatos de vinculación	32
Figura 14	Modal de opciones.....	33
Figura 15	Comunicado Importante	33
Figura 16	Comunicado Encuesta	34
Figura 17	Encuesta interna.....	35
Figura 18	Página para crear un comentario.....	38
Figura 19	Respuesta anidada.....	39
Figura 20	Página para responder a un comentario específico	39
Figura 21	Uso de las herramientas para texto enriquecido	40
Figura 22	Comentarios de un foro actividad.....	40
Figura 23	Hilo de comentarios.....	41
Figura 24	Comentario con menú de opciones limitado.....	42
Figura 25	Listado de docentes pendientes	44
Figura 26	Cajón desplegable de las indicaciones de evaluación.....	45
Figura 27	Cajón desplegable de detalles de la Evaluación	45
Figura 28	Vista de las preguntas del formulario	46
Figura 29	Vista de docentes marcados como evaluados	47
Figura 30	Vista de cuando no hay docentes disponibles a evaluar	47
Figura 31	Vista del menú de opciones de becas.....	50
Figura 32	Vista de solicitud de beca activa.....	50
Figura 33	Vista del formulario de solicitud de beca	51
Figura 34	Vista de subir requisitos activa	51
Figura 35	Vista de información de la solicitud de beca	52
Figura 36	Formulario de requisitos requeridos	53
Figura 37	Vista subir información bancaria activa	53
Figura 38	Vista de información del contrato.....	54
Figura 39	Formulario de información bancaria.....	55
Figura 40	Vista de actividad para evaluación	58
Figura 41	Vista de actividad con solo retroalimentación activa.....	58
Figura 42	Formulario con tipo de pregunta opción única y textual	59
Figura 43	Formulario con tipo de pregunta emparejamiento de columnas	60
Figura 44	Formulario con tipo de pregunta opción múltiple.....	60
Figura 45	Modal de finalización de evaluación	61
Figura 46	Vista del resultado de evaluación	62
Figura 47	Test de normalidad taller práctico	68
Figura 48	Alfa de Cronbach encuesta SUS.....	68
Figura 49	Rango de resultados de calidad en uso	73

INTRODUCCIÓN

Problema de investigación

Hoy en día, acceder a servicios digitales desde el celular se ha vuelto algo completamente habitual [1], y esta tendencia se observa a nivel nacional [2]. Este cambio en la forma en cómo se consumen los servicios impacta especialmente a los usuarios más jóvenes [3], lo que deja claro que cada vez es más importante ofrecer una experiencia de usuario mejorada y optimizada para dispositivos móviles. En este contexto, el servicio "Portafolio Estudiante" de la Universidad Técnica del Norte, implementado en una plataforma web desarrollada en APEX, trae consigo varios retos para los usuarios que necesitan o prefieren usar las funcionalidades académicas desde sus dispositivos móviles. Como se observa en la Figura 1 del trabajo de grado previo [4], la plataforma web no tiene interfaces adaptativas, lo que genera frustración entre los usuarios. Esta limitación en móviles puede terminar afectando la imagen institucional de la Universidad Técnica del Norte, ya que no cumple con los estándares modernos de accesibilidad digital.

En la Fase 1 se cubrieron las funciones esenciales, permitiendo a los usuarios acceder a las principales características del sistema desde la app móvil [4]. Sin embargo, algunas funcionalidades importantes, presentes en la versión web, no fueron implementadas en esa fase. Por esta razón, la Fase 2 busca integrar los módulos pendientes para ampliar el servicio móvil y mejorar de forma notable la experiencia de usuario, la usabilidad y la disponibilidad.



Objetivos

Objetivo general

Implementar la Fase 2 del servicio “Portafolio Estudiante” en la plataforma móvil de la Universidad Técnica del Norte, empleando el marco de trabajo SCRUM y validando su calidad en uso conforme a la norma ISO/IEC 25022.

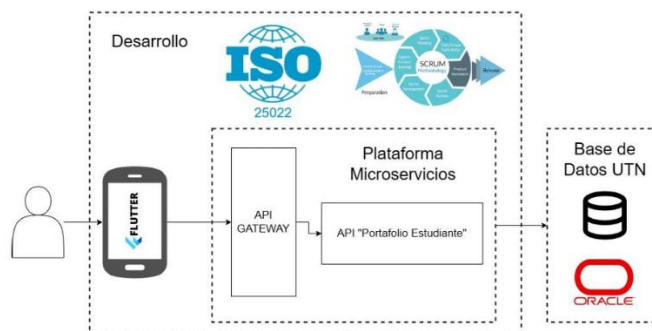
Objetivos específicos

1. Establecer un marco conceptual para el desarrollo de una aplicación móvil de gestión académica.
2. Implementar las nuevas funcionalidades para el servicio del "Portafolio Estudiante" en la aplicación móvil utilizando a través del marco de trabajo SCRUM.
3. Evaluar la calidad de uso de la aplicación móvil para la Fase 2 del servicio “Portafolio Estudiante” con la norma ISO/IEC 25022.

Alcance

El presente trabajo tiene como propósito dar continuidad al desarrollo del servicio "Portafolio Estudiante" en la plataforma móvil de la Universidad Técnica del Norte, implementando módulos no cubiertos en la Fase 1. La segunda fase del proyecto se enfocará en mejorar la experiencia de usuario, añadiendo nuevas funciones que faciliten la gestión académica de los estudiantes dentro de la aplicación móvil. Para garantizar la compatibilidad y escalabilidad del sistema, el proyecto seguirá utilizando la base tecnológica establecida en la primera fase [4], arquitectura limpia con el framework multiplataforma Flutter, permitiendo que la aplicación funcione en Android y en iOS. Además, el backend continuará integrándose con la base de datos Oracle de la universidad y hará uso de las tecnologías establecidas por la Dirección de Desarrollo Tecnológico e Informático (DDTI), asegurando la coherencia con la infraestructura de la universidad, como se muestra en la Figura 2 de [4].

Figura 2 Arquitectura propuesta [4].



En esta fase, se agregarán módulos y funcionalidades que complementen las ya implementadas en la primera versión del servicio. Estas funcionalidades son:

- **Foro Online:** Responder foros para compartir ideas entre estudiantes y docente.
- **Evaluación Online:** Permitirá la realización de pruebas/exámenes en línea.
- **Contenidos de la Materia:** Complementación para más información en el aula virtual de cada materia.
- **Evaluación del Docente:** Permitirá evaluar la calidad de los docentes, con formularios anónimos.
- **Avisos Modales:** Notificaciones que aparecen al ingresar al portafolio estudiante, mostrando algo relevante.
- **Pagos:** Visualización del historial de transacciones que pertenecen a un estudiante.
- **Becas:** Funcionalidad para gestionar el proceso de solicitud de beca.

El desarrollo seguirá el marco de trabajo ágil SCRUM, lo que permitirá una implementación iterativa y flexible, adaptándose a los requerimientos y mejoras necesarias a lo largo del proceso. Además, la evaluación de la calidad del software continuará basándose en la norma ISO/IEC 25022, utilizando métricas de eficacia, eficiencia y satisfacción para medir el impacto de las nuevas funcionalidades en la experiencia del usuario, manteniendo así la coherencia metodológica con la primera Fase [4].

Metodología

Este trabajo de investigación, de carácter aplicado, se centra en la Fase 2 del servicio 'Portafolio Estudiante' dentro de la app móvil de la Universidad Técnica del Norte, este proyecto sigue el trabajo de grado anterior y cumple con las metodologías ya definidas. Para lograr los objetivos, se harán estas actividades principales:

1. Se hará una investigación documental, revisando el proyecto de grado de la Fase 1 [4] para comprender la arquitectura base, y consultando bases de datos científicas que la Universidad Técnica del Norte tiene acceso. Además, usar el gestor bibliográfico Mendeley ayudará a organizar de manera eficiente los trabajos encontrados, permitiendo un manejo más ordenado de la bibliografía.
2. Se aplicará la metodología SCRUM cumpliendo sus tres fases adaptadas (Pre-Juego, Juego, Post-Juego) [5]. En el Pre-Juego se detallarán los requisitos, en el Juego se realizará el desarrollo de las nuevas funcionalidades sobre la arquitectura ya definida en [4], y en el Post-Juego se realizarán las pruebas de aceptación.

Se evaluará la calidad en uso del aplicativo ampliado mediante la norma ISO/IEC 25022 [6], para esto se aplicará una encuesta y un taller práctico con estudiantes para medir eficacia, eficiencia y satisfacción, de acuerdo con el enfoque ya definido [4].

Justificación

Justificación Tecnológica

La continuación del desarrollo del servicio “Portafolio Estudiante” en su segunda fase responde a la necesidad de mejorar la accesibilidad y experiencia del usuario en dispositivos móviles. Esta iniciativa sigue la línea estratégica de la Universidad Técnica del Norte para fortalecer su infraestructura digital, asegurando que los servicios académicos sean más eficientes y adaptados a las tendencias tecnológicas actuales [7].

En la primera fase del proyecto, se establecieron las bases para una plataforma móvil, pero aún quedan módulos por implementar que son esenciales para complementar la funcionalidad del sistema. La segunda fase permitirá incorporar estos módulos, asegurando su integración con la infraestructura institucional existente y optimizando su desempeño en dispositivos móviles. Además, se priorizará la escalabilidad y la calidad del sistema, sentando las bases para futuras mejoras y nuevas fases de desarrollo.

Justificación Social

La expansión del "Portafolio Estudiante" beneficiará directamente a la comunidad universitaria, facilitando la interacción entre estudiantes y la institución mediante herramientas digitales más accesibles y optimizadas.

Este proyecto sigue alineado con el Objetivo de Desarrollo Sostenible (ODS) número 4: "Educación de calidad" [8], garantizando que los estudiantes puedan acceder a información académica de manera eficiente desde cualquier dispositivo.

Contexto de Investigación

Tabla 1 Trabajos Relacionados

INVESTIGACIÓN	APORTE
Contexto: Local Implementó la Fase 1 del servicio "Portafolio Estudiante" en plataforma móvil (UTN), usando Flutter, microservicios, SCRUM y evaluando calidad con ISO/IEC 25022 [4].	Estableció la aplicación móvil base del "Portafolio Estudiante" con funcionalidades esenciales y calidad de uso satisfactoria. El presente trabajo ampliará dicha aplicación implementando módulos académicos.
Contexto: Local Desarrolló una aplicación móvil turística (Android) para el GAD Pedro Moncayo, utilizando Flutter, SCRUM y evaluando usabilidad con ISO/IEC 25010 [9].	El uso de Flutter en la aplicación turística del GAD demuestra que este framework es ideal para crear soluciones móviles robustas. El presente trabajo continuará aplicando esta misma tecnología para desarrollar módulos académicos adicionales en el "Portafolio Estudiante".
Contexto: Nacional Aplicación de un Framework para la evaluación de la calidad en uso del SII-Académico de la Escuela Politécnica Nacional[10].	Demuestra la aplicación práctica de ISO/IEC 25022 en un sistema académico universitario, evaluando usabilidad. El presente trabajo adoptará un enfoque similar de métricas y evaluación para garantizar la calidad en los nuevos módulos.
Contexto: Nacional Creación de Aplicación Web y Mobile para Administración de Reservas de Aulas y Laboratorios para la Universidad Politécnica Salesiana [11].	Desarrollaron un sistema web/móvil para gestionar reservas de espacios en la UPS, usando React para la parte web y Flutter para la parte móvil, además se usó Scrum para mejorar procesos manuales. Este trabajo aplica también Flutter y Scrum, pero en el dominio del Portafolio Estudiante (UTN), sobre una arquitectura de microservicios.
Contexto: Internacional Designing scalable and robust microservice architectures for modern applications [12].	Proporciona estrategias para diseñar microservicios escalables y resilientes. Incluye casos de éxito (Netflix/Amazon) y herramientas (Docker) que validan la escalabilidad requerida para los nuevos módulos.
Contexto: Internacional Microservice-based projects in agile world: A structured interview [13].	Este estudio confirmó que el uso de metodologías ágiles es compatible y común en proyectos con microservicios. Por esta razón valida el enfoque metodológico general del presente trabajo.

I. MARCO TEÓRICO

1.1 Sistema informático integrado universitario (SIIU)

1.1.1 El "portafolio estudiante" en el contexto del SIIU de la UTN

El Sistema Informático Integrado Universitario (SIIU) de la Universidad Técnica del Norte es la plataforma digital central para la gestión de sus operaciones académicas, alineada con la tendencia en educación superior de simplificar procesos y optimizar servicios para los estudiantes. En este entorno, el SIIU incorpora el Portafolio Universitario, concebido como una colección de recursos y actividades seleccionadas cuya finalidad es evidenciar y evaluar periódicamente el rendimiento y aprendizaje, además de potenciar la reflexión y mejora continua de las prácticas académicas, administrativas y profesionales [14].

El "Portafolio Estudiante" ayuda a los alumnos a manejar su historial académico, ya que pueden revisar planes de estudio, ver calificaciones, acceder a materiales y realizar trámites [14]. La primera fase de su versión móvil ya implementó funciones útiles como el calendario académico, acceder a recursos, entregar tareas y consultar horarios [4]. Continuar con su evolución es pertinente, como lo demuestran evaluaciones previas del SIIU en la institución, estas evaluaciones, basadas en el modelo de DeLone y McLean, revelan una percepción favorable de esta herramienta entre los estudiantes [15].

1.1.2 Avance del Servicio "Portafolio Estudiante": De la Fase 1 a los Nuevos Requisitos

La Fase 1 del "Portafolio Estudiante" móvil estableció las bases funcionales del servicio, usando una arquitectura limpia con Flutter en el frontend y una arquitectura de microservicios para el backend, conectada a la base de datos Oracle de la universidad. Las funcionalidades principales se centraron en el calendario, acceso a recursos, entrega de tareas y horarios [4].

La Fase 2 planea una expansión considerable del servicio, la cual busca mejorar la experiencia de los estudiantes universitarios mediante la integración de módulos adicionales como Foro Online, Evaluación Online, Contenidos de la Materia, Evaluación del Docente, Avisos Modales, Pagos y Becas. Este avance es coherente con la tendencia de las aplicaciones móviles educativas, que se vuelven más interactivas y completas [16]. Aunque el desarrollo por fases es una práctica recomendada para gestionar la complejidad

en este tipo de aplicaciones académicas [17], esta diversidad de funciones aumentará notablemente la complejidad interna del microservicio "Portafolio Estudiante", poniendo a prueba su mantenibilidad y requiriendo un control más estricto de la calidad bajo la norma ISO/IEC 25022.

1.1.3 Pertinencia del desarrollo y su alineación con los objetivos institucionales

El proyecto "Portafolio Estudiante" móvil representa una iniciativa importante para la Universidad Técnica del Norte porque muestra su compromiso con la modernización y la calidad académica. La creación de esta aplicación móvil no es solo un avance tecnológico, es también una respuesta a lo que los estudiantes piden: un acceso más fácil y adaptado a su día a día [18].

La iniciativa está reflejada en el Plan Estratégico de Desarrollo Institucional (PEDI) de la Universidad Técnica del Norte, que impulsa la innovación y el fortalecimiento tecnológico de los servicios que esta ofrece. Los proyectos de "Desarrollo de servicios tecnológicos en APPs" muestran este compromiso con la transformación digital. Al mejorar la forma de utilizar el "Portafolio Estudiante" para los estudiantes, la UTN demuestra que se adapta a las necesidades de su comunidad, algo muy importante para formar profesionales en el presente [18].

1.2 Arquitectura de microservicios

1.2.1 Principios de diseño, ventajas y desafíos de los microservicios

La arquitectura de microservicios se basa en principios ya establecidos, cada servicio tiene una función específica, se implementa de manera independiente y funciona con un alto nivel de descentralización en su gestión y manejo de datos [19]. La comunicación entre ellos se realiza mediante APIs RESTful, que utilizan sobre protocolos ligeros como HTTP. Este método ofrece ventajas como una mayor agilidad en el ciclo de vida del desarrollo y una escalabilidad más precisa, ya que solo se amplían los servicios que lo requieren [20]. Esta arquitectura también aporta una alta resiliencia ante los fallos que se presenten y permite el uso de diversas tecnologías entre servicios. Sin embargo, también presenta sus propios desafíos, como una complejidad operativa superior, posible latencia en la comunicación entre los servicios y dificultades para mantener la consistencia de los datos distribuidos [21].

1.2.2 Desarrollo backend con NestJS y Typescript para microservicios

Para la construcción del backend se seleccionaron tecnologías modernas que aseguran una base de código mantenible y escalable a lo largo del tiempo.

- **TypeScript.** Es un superconjunto de JavaScript que ha sido desarrollado por Microsoft que añade un sistema de tipado estático opcional, esta característica es clave para la detección temprana de errores durante la fase de desarrollo, lo que mejora la calidad y robustez del código, en especial para proyectos grandes. Al compilarse a JavaScript estándar, funciona en cualquier parte, pero facilita la función del desarrollador con interfaces, decoradores y una mejor visión del código [22], [23].
- **NestJS.** Es un framework de Node.js que ha ganado popularidad rápidamente y cuya arquitectura modular similar a la de Angular, promueve una clara separación de responsabilidades, lo que hizo que fuera elegido para este proyecto por su capacidad para crear aplicaciones complejas y de alto rendimiento, resulta fácil de mantener y escalar, además de que ofrece soporte nativo para diversos transportadores de comunicación (TCP, NATS, Kafka), lo cual es muy valioso en un entorno de microservicios [24].

1.2.3 Interfaces de programación de aplicaciones (APIs REST) y Comunicación

La comunicación en una arquitectura de microservicios se maneja mediante APIs que actúan como acuerdos entre componentes, siendo el estilo arquitectónico REST el enfoque dominante por su simplicidad y porque no guarda estado. Para manejar eficientemente este flujo de comunicación entre el cliente móvil y el backend, se utiliza un API Gateway, este patrón centraliza las solicitudes externas en un solo punto [25]. Este componente no solo redirige el tráfico hacia el servicio interno correcto, sino que también libera a los microservicios de tareas comunes como la autenticación y la autorización, aunque su gestión debe ser cuidadosa para evitar que se convierta en un punto de fallo central [25], [26].

1.2.4 Integración de microservicios con bases de datos (Oracle)

Mientras la teoría de microservicios sugiere tener una base de datos por cada servicio [27], en entornos institucionales como el de la UTN es necesario adoptar un enfoque más práctico y realista, ya que este proyecto sigue la arquitectura de la Fase 1,

que está diseñada para operar sobre la base de datos Oracle centralizada de la universidad, entonces esto significa que la separación de datos debe manejarse a nivel lógico [4].

- **Patrón de Repositorio.** Para gestionar esta interacción de manera ordenada y eficiente, se implementa este patrón como una capa de abstracción. Esta capa actúa como intermediario entre la lógica de negocio y la fuente de datos, ocultando los detalles de cómo se consultan y almacenan los datos en la base de datos de Oracle [20], [28].
- **TypeORM.** Es la herramienta que implementa el patrón de repositorio en este proyecto, ya que se trata de un Mapeador Objeto-Relacional (ORM) que maneja la conexión entre NestJS y la base de datos Oracle. Su función principal es convertir el modelo relacional de la base de datos a los objetos con los que trabaja la aplicación en TypeScript, lo que asegura consistencia con la arquitectura existente y mejora la productividad del desarrollo [29].

1.3 Aplicaciones móviles

1.3.1 Desarrollo multiplataforma con Flutter y lenguaje Dart

El desarrollo de aplicaciones móviles se puede enfocarse desde diferentes puntos, como el nativo, el web o el multiplataforma móvil [30], este último ha ganado mucha popularidad por su capacidad para reutilizar una única base de código para dos sistemas operativos diferentes, lo que reduce los tiempos y costos de desarrollo mientras permite llegar a más usuarios. Teniendo eso en cuenta, Flutter, un kit de desarrollo de software (SDK) de código abierto creado por Google, se ha establecido como uno de los frameworks de más importantes para el desarrollo multiplataforma [31].

- **Lenguaje Dart.** Flutter utiliza el lenguaje de programación Dart, también creado por Google, que está optimizado para la creación de interfaces de usuario. Una de las características más destacadas de Dart es la compilación AOT que permite un rendimiento rápido y nativo, esencial para la experiencia del usuario final y, por otro lado, la compilación JIT facilita un desarrollo ágil, gracias a la característica de "hot reload", esta funcionalidad es particularmente útil, ya que permite a los desarrolladores ver los cambios en el código reflejados casi al instante en la aplicación, lo que acelera significativamente el desarrollo [32], [33], [34].

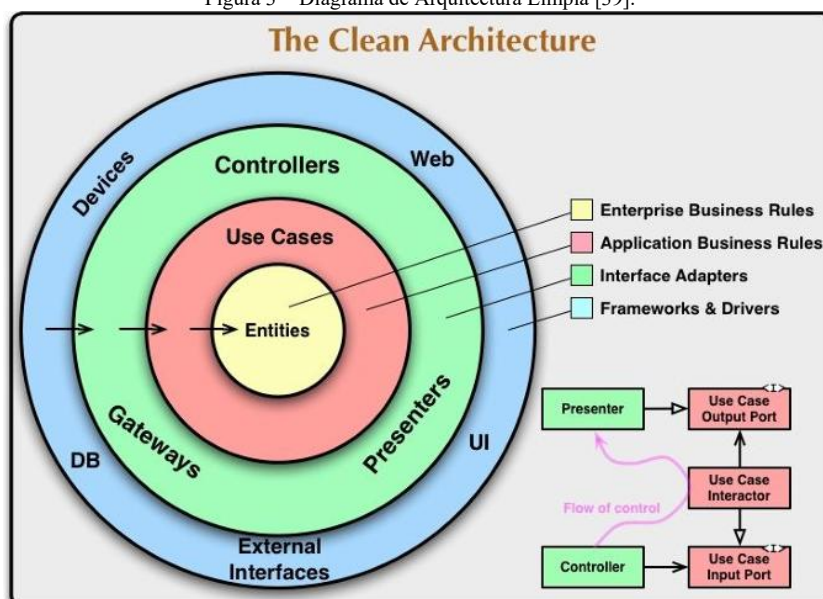
- **Arquitectura basada en Widgets.** La arquitectura de Flutter está basada en el concepto de widgets, que son los elementos esenciales para la construcción de toda la interfaz de usuario [35], ya que en Flutter todo es un widget, desde un simple botón hasta diseños complejos, lo que permite una gran flexibilidad sobre la apariencia y el comportamiento de la aplicación [36].

La elección de Flutter en proyectos institucionales, como los desarrollados en la UTN, demuestra una preferencia por sus ventajas en el desarrollo ágil y la capacidad de ofrecer experiencias de usuario similares en distintas plataformas. Aunque Flutter alcanza un rendimiento cercano al nativo, la calidad final de las aplicaciones también depende de buenas prácticas de programación, una gestión del estado correcta y una comunicación eficiente con los servicios del backend [37].

1.3.2 Arquitectura limpia en aplicaciones Flutter

Para desarrollar aplicaciones que sean mantenibles y escalables, es importante seguir patrones arquitectónicos ya definidos, la arquitectura Limpia, popularizada por Robert C. Martin [38], es un conjunto de principios de diseño que promueve la separación de responsabilidades con el objetivo de independizar la lógica de negocio de los frameworks, la interfaz de usuario y las fuentes de datos. Su regla principal es que las dependencias solo puedan fluir desde las capas externas hacia las internas, así obteniendo el modelo que propone una separación en cuatro capas principales, como se muestra en la Figura 3 [39].

Figura 3 Diagrama de Arquitectura Limpia [39].



Para entender cómo funcionan estas capas, podemos desglosarlas de la siguiente manera:

- **Capa de Entidades (Entities):** Es el círculo más interno y contiene los objetos y las reglas de negocio más generales y de alto nivel de la aplicación. Es la parte más estable y la que menos cambios sufre con el tiempo, porque las reglas de negocio suelen ser consistentes y duraderas [39].
- **Capa de Casos de Uso (Use Cases):** Contiene la lógica de negocio específica de la aplicación, ya que esta maneja el flujo de datos desde y hacia la capa de Entidades y dirige a los adaptadores de interfaz para que cumplan los objetivos del caso de uso [39].
- **Capa de Adaptadores de Interfaz (Interface Adapters):** Este nivel actúa como un conjunto de convertidores que adaptan los datos a un formato conveniente tanto para los casos de uso y entidades como para las capas más externas (la base de datos o la UI). Aquí se encuentran componentes como los Presenters y Controllers [39].
- **Capa de Frameworks y Dispositivos (Frameworks & Drivers):** Es la capa más propensa a cambios, ya que contiene los detalles específicos de las herramientas, como el framework de la interfaz de usuario, los dispositivos y el acceso a la base de datos [39].

La adopción de este patrón en el proyecto "Portafolio Estudiante" ofrece muchos beneficios en términos de testabilidad y mantenibilidad. Pero no todo es bueno ya que puede introducir una mayor cantidad de código en proyectos pequeños, sus ventajas se hacen visibles a medida que la aplicación crece en complejidad, garantizando su adaptabilidad a largo plazo [38]. Por esta razón es clave es la "regla de dependencia", que asegura que la lógica de negocio central permanezca independiente de las tecnologías externas, ya que esta independencia permite que la aplicación evolucione [39].

1.3.3 Desarrollo iterativo y evolución funcional de aplicaciones móviles

El desarrollo de aplicaciones móviles, por su naturaleza interactiva y el rápido cambio tecnológico, se beneficia de un enfoque iterativo e incremental [30]. Este método, clave en las metodologías ágiles como Scrum, consiste en construir la aplicación en ciclos

cortos, donde cada ciclo, o iteración, produce una versión funcional del software que aporta valor y puede ser evaluada por los usuarios y partes interesadas [40].

El desarrollo de una aplicación móvil no sigue una línea recta, sino un ciclo constante de desarrollo, prueba, retroalimentación y adaptación [41]. Las primeras iteraciones pueden centrarse en las funcionalidades principales, y las iteraciones siguientes mejoran estas funcionalidades y añaden nuevas características basadas en las necesidades emergentes y comentarios recibidos [40]. Esta capacidad de "responder al cambio por encima de seguir un plan" es uno de los valores fundamentales del Manifiesto Ágil y es útil en el desarrollo móvil, donde las expectativas de los usuarios y las capacidades de los dispositivos evolucionan rápidamente [42].

La gestión de requisitos cambiantes y el "feature creep" son desafíos comunes en el desarrollo de aplicaciones [43]. Para enfrentarlos, es importante tener un proceso iterativo bien estructurado, con una priorización clara del backlog y una comunicación frecuente con el propietario del producto [40]. Además, la gestión de versiones y un plan de lanzamientos definido son esenciales para controlar las mejoras y evolución de la aplicación y garantizar que las nuevas versiones lleguen a los usuarios con una calidad adecuada [44]. En el caso del "Portafolio Estudiante" este enfoque permite que la herramienta se adapte de manera continua a las necesidades académicas y de usabilidad de los estudiantes.

1.3.4 Consideraciones de seguridad y privacidad en aplicaciones móviles

La seguridad y la privacidad son pilares fundamentales en el desarrollo de aplicaciones móviles, especialmente cuando se maneja datos sensibles, como en el caso de esta aplicación, que maneja datos académicos y personales. Entre las amenazas más comunes se encuentran el almacenamiento inseguro de datos en el dispositivo, la comunicación insegura con los servidores backend, la inyección de código malicioso, y el uso inadecuado de las funcionalidades de la plataforma. La guía OWASP Mobile Top 10 ofrece un marco de referencia valioso para identificar y mitigar estos riesgos de seguridad más comunes en aplicaciones móviles [45].

Para aplicaciones Flutter, las consideraciones de seguridad específicas incluyen:

- **Almacenamiento seguro de datos:** Emplear mecanismos de almacenamiento cifrado para datos sensibles como tokens de autenticación. Paquetes como

flutter_secure_storage recurren a las capacidades nativas de iOS (Keychain) y Android (Keystore) para este fin [46].

- **Comunicación segura:** Implementar el protocolo HTTPS en todas las comunicaciones de red y también se podría considerar técnicas como el "certificate pinning" que permiten prevenir ataques de intermediario [47].
- **Validación de entradas:** Validar y asegurar todas las entradas del usuario para prevenir ataques de inyección [48].
- **Protección del binario:** Aplicar técnicas de ofuscación de código para dificultar la ingeniería inversa de la aplicación [49].
- **Gestión de permisos:** Solicitar los permisos que realmente sean necesarios para el funcionamiento de la aplicación y así evitar accesos innecesarios que puedan comprometer la seguridad.

En cuanto a la privacidad, es importante seguir los principios como la minimización de datos, lo que implica recolectar y almacenar solo la información necesaria, obtener el consentimiento informado del usuario para el tratamiento de sus datos, y cumplir con las regulaciones de protección de datos. La seguridad móvil es una responsabilidad compartida entre la aplicación cliente el frontend y los servicios backend, ya que las vulnerabilidades en un extremo pueden comprometer la seguridad del otro. Por lo cual, un enfoque de seguridad completa es esencial para proteger la información de los estudiantes y mantener la confianza en la aplicación [45], [50].

1.3.5 Estrategias para la gestión de errores y resiliencia en aplicaciones móviles

Una gestión de errores robusta y la construcción de aplicaciones resilientes son cruciales para ofrecer una experiencia de usuario positiva y mantener la estabilidad de las aplicaciones móviles. Los errores pueden surgir de diversas fuentes: problemas de red, fallos en las APIs del backend, errores en el manejo de datos locales, o problemas en la propia lógica de la interfaz de usuario. En Flutter, la gestión de errores se puede implementar utilizando bloques try-catch para código síncrono y asíncrono (Futures y Streams), y es importante presentar al usuario mensajes de error claros y comprensibles en lugar de fallos abruptos [51].

Desarrollar una aplicación resiliente requiere planificar cómo responder ante situaciones inesperadas:

- **Manejo de la conectividad:** La aplicación debe detectar cuando no hay internet y seguir funcionando mediante datos almacenados localmente.
- **Degradación controlada:** Si un servicio falla, la aplicación ofrece funciones básicas en lugar de dejar de funcionar completamente.
- **Gestión de estado:** Se utilizan patrones que permiten recuperar la información del usuario después de una interrupción.
- **Reintentos inteligentes:** Cuando ocurre un error temporal de red, el sistema reintenta la conexión de forma automática.

Además de manejar errores internos, es necesario contar con herramientas de monitoreo para detectar problemas en producción. Servicios como New Relic para Flutter [52] u otras herramientas similares permiten rastrear errores de código, solicitudes de red y otros eventos. Esta información resulta valiosa para mantener y mejorar la aplicación de forma continua.

1.4 Metodología de desarrollo de software

1.4.1 Fundamentos del Desarrollo Ágil y el Manifiesto Ágil

El desarrollo ágil de software apareció para solucionar los problemas de los métodos tradicionales, que tenían ciclos de desarrollo muy largos y les costaba mucho adaptarse a los cambios. Así que, en 2001, un grupo de expertos escribió el "Manifiesto por el Desarrollo Ágil de Software" para definir estos nuevos enfoques. Este texto se basa en cuatro valores principales [53], [54]:

1. Dar más importancia a las personas y su interacción que a los procesos establecidos.
2. Desarrollar software que funcione en vez de crear mucha documentación.
3. Priorizar la colaboración con el cliente más que negociar contratos.
4. Estar preparados para cambiar el rumbo en vez de seguir un plan fijo.

Estos valores se justifican en doce principios que guían la práctica ágil, algunos de ellos destacan la satisfacción del cliente a través de entregas tempranas y continuas, la

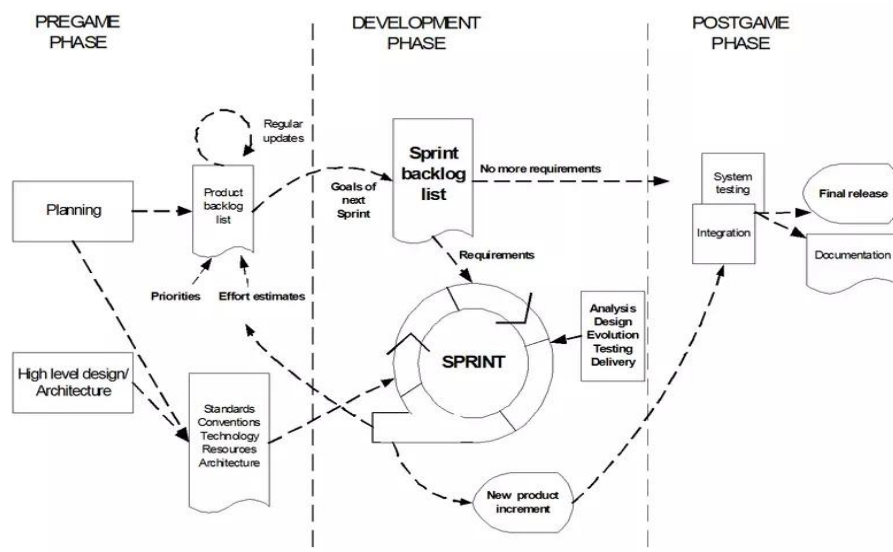
aceptación de requisitos cambiantes, la colaboración diaria entre negocio y desarrolladores, y la construcción de proyectos en torno a individuos motivados [55]. Es crucial entender que "ágil" es más una mentalidad que una metodología prescriptiva [56].

1.4.2 El Marco de Trabajo Scrum y su Proceso de Desarrollo

Scrum es una forma sencilla de trabajar que ayuda a los equipos a crear valor a través de soluciones flexibles para problemas difíciles, basada en la experiencia que se refleja en tres pilares: transparencia, revisión y adaptación, con una estructura compuesta por roles, eventos y herramientas que se comprenden mejor cuando se aplican durante el desarrollo de un producto [40].

Aunque la Guía Scrum se centra en el ciclo de Sprints, el desarrollo de un producto puede entenderse dentro de un marco más amplio basado en el trabajo de Takeuchi y Nonaka que son autores que contribuyeron a la creación de Scrum, donde el proceso se divide en tres fases principales, como se muestra en la Figura 4 [57] y que también fue aplicado durante la Fase 1 de este proyecto [4].

Figura 4 Fases del Proceso de Desarrollo [57].



- Fase de Pre-Juego (Pre-Game): Esta fase inicial se centra en la planificación y el diseño de alto nivel. Se divide en dos sub-fases [57]:
 - Planificación: Se define qué sistema se va a desarrollar y se crea una lista inicial llamada Product Backlog con todo lo que se sabe que se necesita, luego se ordenan estas necesidades por importancia y se

estima el esfuerzo requerido para cada una mientras se elige el equipo y las herramientas [57].

- Arquitectura: Se diseña la estructura principal del sistema usando lo que hay en el Product Backlog, si el sistema ya existe se ve qué cambios hay que hacer y qué problemas podrían surgir.
- Fase de Juego (Game): Aquí es donde se desarrolla el producto paso a paso a través de varios Sprints. Cada Sprint dura de una a cuatro semanas y en ese tiempo se crea una nueva parte del producto. En esta fase, el equipo usa eventos y herramientas importantes [57]:
 - Sprint Planning Meeting: El equipo, junto al Product Owner y otros interesados, decide los objetivos y la funcionalidad del siguiente Sprint [57].
 - Sprint Backlog: Se crea una lista de los elementos del Product Backlog seleccionados para ser implementados durante el Sprint y a diferencia del Product Backlog, esta lista permanece estable durante el ciclo [57].
 - Daily Scrum Meeting: Reuniones rápidas de unos 15 minutos cada día para que el equipo esté alineado, ver cómo va todo y planear el trabajo del día siguiente [57].
 - Sprint Review Meeting: Al final del Sprint, el equipo presenta el incremento de producto funcional a los interesados para evaluar el resultado y decidir los siguientes pasos [57].
- Fase de Post-Juego (Post-Game): Esta fase se empieza cuando se determina que los requisitos del producto están completos y no se añadirán nuevos elementos. El sistema está listo para salir al mercado, así que ahora toca prepararlo: hacer la integración final, pruebas completas y la documentación [57].

1.4.3 Aplicación de Scrum en proyectos de desarrollo de software

Scrum se ha convertido en uno de los marcos ágiles más usados para desarrollar software en todo el mundo. Su popularidad se explica por su capacidad para manejar proyectos complicados y entregar resultados de forma gradual y adaptable. Este marco se

aplica en diferentes tipos de empresas, desde startups pequeñas hasta compañías grandes, y funciona bien para crear aplicaciones web, móviles y sistemas empresariales [40].

Scrum ofrece muchas ventajas, ya que facilita el trabajo colaborativo y mejora la comunicación entre el equipo y las personas involucradas mientras permite una adaptación rápida a los cambios solicitados por clientes o el mercado, promueve entregas constantes desde etapas tempranas que ayudan a recibir opiniones y mejorar el producto de forma continua, hace más manejables los riesgos al hacerlos visibles y resolverlos progresivamente, y fomenta la productividad y motivación del equipo al permitirles organizarse por sí mismos y tomar decisiones propias. Implementar Scrum no siempre es sencillo, ya que muchas empresas tradicionales se resisten a cambiar su forma de trabajar, lo que representa un obstáculo, y también ocurre que algunos equipos lo aplican de forma incompleta siguiendo solo ciertas partes del marco lo que se conoce como "Scrum-but", por eso, más que seguir reglas al pie de la letra, lo esencial es comprender y aplicar sus valores fundamentales: transparencia, revisión y adaptación, ya que Scrum funciona como una guía flexible que los equipos deben adaptar a su contexto para que realmente sea eficaz [40].

1.5 Calidad del software y modelos de evaluación

1.5.1 La Familia de normas ISO/IEC 25000 (SQuaRE)

La familia de normas ISO/IEC 25000, también conocida como SQuaRE (System and Software Quality Requirements and Evaluation), constituye el estándar internacional de referencia para la evaluación de la calidad del producto de software. Esta serie de normas unifica y reemplaza estándares anteriores como ISO/IEC 9126 (enfocada en el modelo de calidad) e ISO/IEC 14598 (enfocada en el proceso de evaluación), proporcionando un marco de trabajo completo para la especificación y evaluación de la calidad [58].

SQuaRE se estructura en varias partes, cada una enfocada en un aspecto concreto que se mantiene presente desde el desarrollo hasta el uso.

Figura 5 Divisiones de la familia de normas ISO/IEC 25000 [58].



- **ISO/IEC 2500n – División de Gestión de la Calidad:** Establece los modelos, términos y definiciones comunes para la gestión de la evaluación de la calidad [58].
- **ISO/IEC 2501n – División del Modelo de Calidad:** Presenta los modelos de calidad detallados, incluyendo los de calidad de producto, calidad en uso y calidad de datos [58].
- **ISO/IEC 2502n – División de Medición de la Calidad:** Provee un modelo de referencia y definiciones de métricas para cuantificar la calidad de forma objetiva [58].
- **ISO/IEC 2503n – División de Requisitos de Calidad:** Ofrece guías para la especificación de requisitos de calidad del software [58].
- **ISO/IEC 2504n – División de Evaluación de la Calidad:** Proporciona requisitos y recomendaciones para conducir el proceso de evaluación [58].

1.5.2 Modelo de Calidad de la Norma ISO/IEC 25010

Dentro de SQuaRE, la norma ISO/IEC 25010 es clave porque define los modelos para clasificar la calidad del software y presenta dos enfoques principales: uno orientado a la calidad del producto, que analiza características como confiabilidad, seguridad y facilidad de mantenimiento, y otro centrado en la calidad en uso, que resulta especialmente relevante para este proyecto ya que se enfoca en la experiencia de los estudiantes al utilizar la aplicación [59].

1.5.3 Calidad en uso

En el marco SQuaRE, la Calidad en Uso resulta clave para este proyecto porque, a diferencia de la calidad del producto que se enfoca en las características del software, este modelo evalúa qué tan bien los usuarios logran sus objetivos al interactuar con el sistema, considerando si lo hacen de forma efectiva, eficiente y satisfactoria dentro de un contexto específico, lo que permite medir la calidad del software desde la perspectiva directa de la experiencia del usuario [60].

1.5.4 Norma ISO/IEC 25022: Medición de la Calidad en Uso

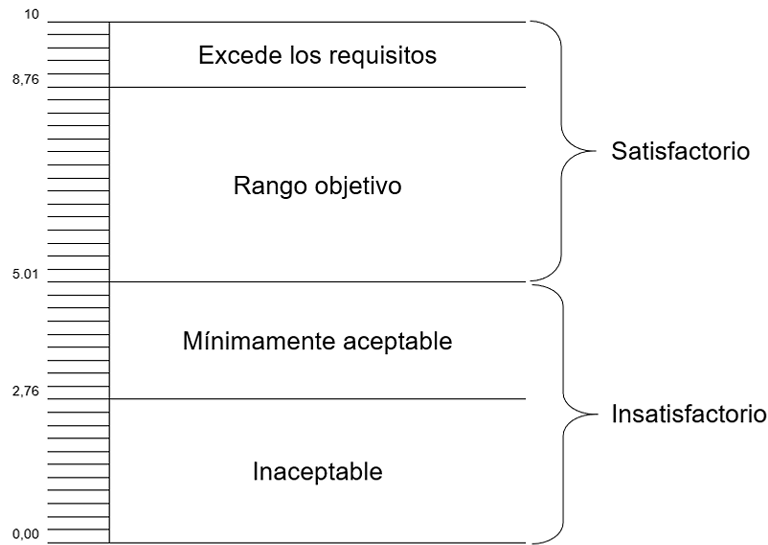
Para medir la Calidad en Uso, este proyecto usa la norma ISO/IEC 25022 [61]. Esta norma dice qué métricas se necesitan para evaluar las características de la experiencia del usuario que define la ISO/IEC 25010. Como ya se hizo en la Fase 1 [4], la evaluación de la aplicación móvil se enfocará en tres características:

- **Eficacia:** Mide si el sistema ayuda a los usuarios a lograr sus objetivos de manera correcta y completa en un contexto específico. Evalúa si las tareas se pueden completar bien y si se obtienen los resultados esperados.
- **Eficiencia:** Aquí se mide cuánto tiempo y esfuerzo mental necesita el usuario para cumplir sus objetivos sin desperdiciar recursos innecesariamente.
- **Satisfacción:** Se trata de cómo se siente el usuario al usar el sistema y si realmente cubre sus necesidades. Aquí entran en juego cosas como qué tan útil le parece, si confía en el sistema y lo cómodo que se siente al usarlo.

1.5.5 Escala de Medición de la Calidad en Uso

Para interpretar los resultados numéricos de las métricas es necesario emplear una escala de medición estándar, por eso en este proyecto se utiliza la propuesta por la norma ISO/IEC 25040 [62], la cual permite clasificar el nivel de calidad en categorías específicas como se muestra en la Figura 6, donde los resultados se organizan en dos rangos principales: Satisfactorio e Insatisfactorio, cada uno con sus respectivas subcategorías.

Figura 6 Escala de medición de Calidad en Uso [62].



1.6 Trabajos relacionados

En [4], se propuso la implementación de la Fase 1 del servicio "Portafolio Estudiante" en la plataforma móvil para la Universidad Técnica del Norte, con el objetivo de solucionar los problemas de accesibilidad y experiencia de usuario de la plataforma web existente. Para lograrlo, se creó una aplicación multiplataforma usando el framework Flutter con arquitectura limpia, que se conecta con una plataforma de microservicios hecha en NestJS. El proyecto se gestionó con la metodología SCRUM y se verificó la calidad en uso según la norma ISO/IEC 25022, mediante talleres y encuestas con estudiantes. El dato más importante fue la calificación de 82.39% en calidad de uso, lo que pone a la aplicación en un rango "satisfactorio" y muestra que la solución funciona bien al incluir herramientas básicas como ver notas, horarios y entregar tareas. Sin embargo, el estudio tuvo la limitación de no agregar módulos académicos importantes que existen en la versión web, como los foros y las evaluaciones en línea, lo que hace que los usuarios deban usar la plataforma web para esas actividades y muestra la necesidad de una segunda fase de desarrollo para tener las mismas funciones en ambas versiones.

El trabajo de Enríquez Rosero [63] buscó implementar el servicio "Portafolio Docentes" en la plataforma móvil de la Universidad Técnica del Norte, para actualizar el sistema existente en Oracle APEX y resolver sus problemas de acceso y facilidad de uso. Se usó el marco de trabajo Scrum para gestionar el proyecto, creando una aplicación multiplataforma con Flutter y una arquitectura de microservicios con NestJS para el

backend. La calidad se midió según la norma ISO/IEC 25022, usando talleres prácticos y encuestas SUS con 22 docentes. El dato más importante fue un índice de calidad de uso del 87.63%, que califica la aplicación como "satisfactoria" y muestra buenos resultados en las pruebas. Como limitación, el estudio encontró diferencias con el portal web y no incluyó funciones complejas como crear o responder foros como docente o crear evaluaciones en línea con su banco de preguntas, haciendo que el usuario deba usar la versión web para esas actividades.

En [9], se trabajó el problema de la poca promoción turística en el cantón Pedro Moncayo creando una aplicación móvil para dar a conocer sus lugares y eventos importantes. Para lograrlo, el proyecto se manejó con el marco de trabajo Scrum; se hizo una aplicación móvil para Android con el framework Flutter y una página web de administración con ReactJs. El backend se construyó como una API REST monolítica usando Node.js y una base de datos MongoDB. La calidad del software se midió evaluando la usabilidad según la norma ISO/IEC 25010, aplicando una encuesta SUS a 32 personas. El dato más importante fue que la aplicación móvil obtuvo una puntuación promedio de 81.79 en la escala SUS, lo que significa un grado de aceptación 'A' ("Excelente") por parte de los usuarios. Sin embargo, el proyecto tuvo dos limitaciones principales: la aplicación se creó solo para Android, dejando fuera a los usuarios de iOS, y el backend usó una arquitectura monolítica, sin considerar opciones más modulares y escalables como los microservicios.

En [64] se muestra un caso de estudio sobre el desarrollo de "IscteSpots", una aplicación gamificada para difundir la historia y el patrimonio de la universidad Iscte en Portugal. El objetivo fue probar la viabilidad y ventajas de usar un framework multiplataforma en un proyecto real y completo. Para esto, se usó el framework Flutter con Dart para crear una única base de código desplegada en la web, Android y iOS. El backend se hizo con un servidor Django que ofrecía una API REST y un portal de gestión de contenidos. El desarrollo fue iterativo, usando modelos BPMN y UML para la planificación y una tubería de CI/CD con CodeMagic y SonarCloud para asegurar la calidad del código. La validación se hizo con 22 estudiantes usando los instrumentos NASA-TLX y MARS. El resultado principal fue el despliegue exitoso en las tres plataformas con mucha reutilización de código (solo el 14.8% del código necesitó cambios para la versión web) y con una respuesta muy positiva de los usuarios en las pruebas de usabilidad. La principal limitación técnica fue la incompatibilidad de algunos

paquetes externos (ej. sqlite) entre las plataformas móviles y la web, lo que requirió reescribir ciertas partes y adaptar la API.

Wang *et al.* [65] presentan la investigación y desarrollo de una aplicación de servicios integrales para estudiantes universitarios en China, con el objetivo de solucionar la dispersión de servicios y la mala experiencia de usuario de las aplicaciones de campus actuales. Para esto, el sistema se diseñó sobre una arquitectura de microservicios usando el framework Spring Cloud Alibaba, mientras que la aplicación móvil se hizo con Flutter para asegurar compatibilidad multiplataforma y buen rendimiento. El backend usa varias técnicas de optimización, como un sistema de caché de cuatro capas y procesamiento asíncrono, para manejar muchos usuarios al mismo tiempo. El resultado es el diseño de una plataforma "todo en uno" que incluye una comunidad virtual, servicios de comercios locales, gestión de eventos y navegación, entre otras funciones. La principal limitación del estudio es que se centra en describir la arquitectura y las funciones, pero no incluye pruebas con usuarios reales para medir de forma concreta la usabilidad o el impacto verdadero de la aplicación en la comunidad estudiantil.

Ünlü *et al.* [13], con el objetivo de comprender las prácticas industriales actuales, investigan cómo las organizaciones que adoptan la arquitectura de software basada en microservicios (MSSA) gestionan sus proyectos dentro de un marco de trabajo ágil. La metodología empleada fue una serie de entrevistas estructuradas con 20 profesionales de diversas organizaciones, roles y países. Se indagó sobre sus procesos de análisis, diseño, pruebas, medición de tamaño y estimación de esfuerzo. El resultado principal revela que las organizaciones consideran que el desarrollo ágil (principalmente Scrum) es altamente compatible con los microservicios. Sin embargo, tienden a reutilizar métodos y notaciones tradicionales de la orientación a objetos para el análisis y diseño, y continúan empleando enfoques de estimación subjetivos como Story Points. La principal limitación que enfrentan estas organizaciones son los desafíos únicos para probar las dependencias entre servicios y la falta de herramientas ágiles que cubran de forma integral todo el ciclo de vida de los microservicios.

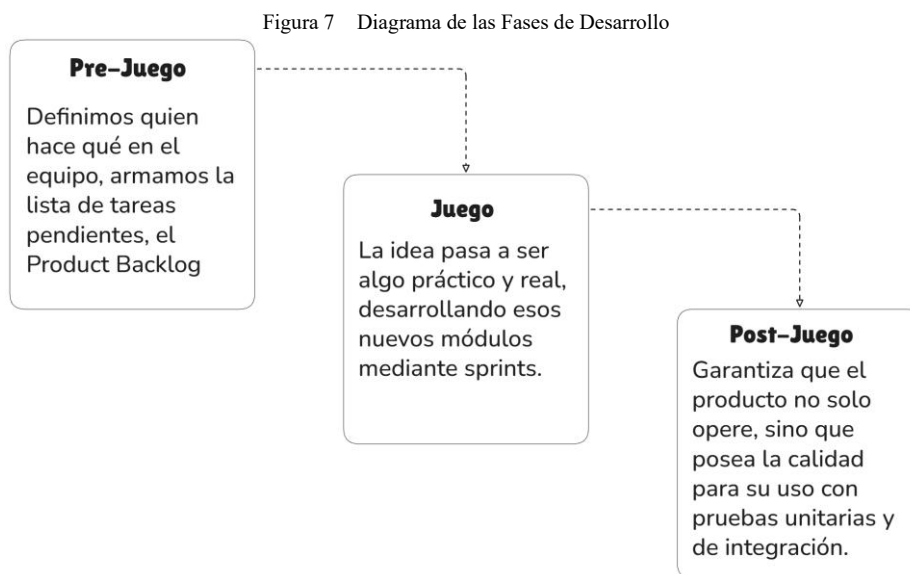
Reddy *et al.* [66] propusieron el desarrollo de la aplicación móvil "ProfBook" con el objetivo de solucionar la ineficiente gestión manual para la reserva de espacios académicos (aulas, auditorios) en una institución de educación superior. Para su implementación, utilizaron un enfoque de Design Thinking, desarrollando la aplicación multiplataforma con el framework Flutter y empleando Firebase como solución integral

para el backend, incluyendo la base de datos (Firestore) y la autenticación de usuarios. El resultado más relevante fue la creación de una plataforma funcional y centralizada que simplifica el proceso de reserva, ofreciendo a los docentes una interfaz para solicitar lugares y a los administradores un sistema para gestionar dichas solicitudes, evitando conflictos de programación. No obstante, el estudio presenta como limitación la ausencia de una validación formal de la calidad en uso mediante estándares como la norma ISO y su arquitectura basada en Firebase, si bien facilita un desarrollo rápido, puede ofrecer menos control y escalabilidad a largo plazo en comparación con una arquitectura de microservicios personalizada.

Maldonado Pila [10] tuvo como objetivo evaluar la calidad en uso del Sistema Integrado de Información Académico (SII-Académico) de la Escuela Politécnica Nacional, con el fin de identificar áreas de mejora y comprender la percepción de los usuarios. Para ello, aplicó un framework de evaluación basado en la familia de normas ISO/IEC 25000, utilizando el proceso de la ISO/IEC 25040 y las métricas de la ISO/IEC 25022 para medir la efectividad, eficiencia y satisfacción a través de talleres prácticos y encuestas con docentes y estudiantes. El resultado más relevante fue la obtención de una calificación global de calidad en uso de 8.46 sobre 10, ubicando al sistema en un grado de satisfacción "aceptable". La principal limitación identificada en el sistema evaluado fue una baja eficiencia (2.32/3), ya que los usuarios nuevos tardaban más que los expertos en completar las tareas, además de reportar insatisfacción con la claridad de los mensajes y la complejidad de la navegación.

II. DESARROLLO

En este capítulo se detallan las fases de ejecución del proyecto bajo la metodología ágil Scrum. El proceso se divide en tres fases principales: Pre-Juego, Juego y Post-Juego como se muestra en la Figura 7. Esta estructura permite que el desarrollo del proyecto tenga una guía de principio a fin, desde la elicitación de requisitos hasta la entrega de un producto funcional.



2.1 DESARROLLO FASE - PRE-JUEGO

Durante esta etapa se definieron los roles de gestión, se realizó la elicitación de requisitos mediante historias de usuario y se estableció el orden de prioridad en el Product Backlog.

2.1.1 Asignación de Roles del Equipo Scrum

En la Tabla 2 se detalla la como se compone el equipo de trabajo.

Tabla 2 Asignación de Roles

Nombre	Rol Scrum	Cargo
Antonio Quiña	Product Owner	Director de Trabajo de Integración Curricular (Docente UTN)
Jordan Puruncajas Castillo	Scrum Master	Tesisista
	Team Development	

Dirección de Desarrollo Tecnológico e Informático	Stakeholder	Departamento TI de la Institución
---	-------------	-----------------------------------

2.1.2 Definición de Requisitos

La obtención de estos requisitos se realizó bajo el formato estándar de historias de usuario. Este enfoque permite centrar el desarrollo en el valor que recibe el estudiante, facilitando la validación de los criterios de aceptación en etapas posteriores. A continuación, se presentan todas las historias de usuario para esta fase.

Tabla 3 Historia de Usuario 1: Pagos

Código: HU_01	Título: Visualización del Historial de Pagos	Prioridad: Baja
Descripción:	Como estudiante, quiero visualizar mi historial completo de transacciones financieras con la universidad.	
Criterios de Aceptación:	CA1: El sistema debe listar todas las transacciones históricas del estudiante, ordenadas descendientemente por fecha. CA2: Cada ítem de la lista debe mostrar: Concepto, Monto, Fecha y Estado. CA3: Si la transacción dispone de archivos asociados (factura electrónica o comprobante de matrícula), se debe habilitar un botón para poder acceder a descargar este documento.	

Tabla 4 Historia de Usuario 2: Contenidos de la Materia

Código: HU_02	Título: Árbol de Contenidos de la Materia	Prioridad: Media
Descripción:	Como estudiante, quiero visualizar la estructura de temas de cada unidad de mis asignaturas.	
Criterios de Aceptación:	CA1: Al ingresar a esta sección, se debe mostrar una vista de árbol o lista expandible que represente las Unidades y sus respectivos Temas. CA2: La estructura debe reflejar fielmente la planificación curricular.	

Tabla 5 Historia de Usuario 3: Formatos de Vinculación

Código: HU_03	Título: Descarga de Formatos de Vinculación	Prioridad: Baja
Descripción:	Como Estudiante, quiero acceder a una página dedicada con la lista de formatos de vinculación (archivos Excel y Word) para poder descargarlos en mi dispositivo, o abrirlos con una aplicación compatible.	
Criterios de Aceptación:	CA1: El sistema debe mostrar una lista clara de archivos disponibles.	
	CA2: Cada ítem de la lista debe mostrar el nombre del archivo y un icono representativo de su tipo (Word o Excel).	
	CA3: Al hacer presionar en un archivo, este debe dejar elegir si lo quiere descargar automáticamente al dispositivo, o la opción para abrir con una aplicación compatible.	

Tabla 6 Historia de Usuario 4: Avisos Modales

Código: HU_04	Título: Sistema de Avisos Modales	Prioridad: Alta
Descripción:	Como estudiante, quiero recibir comunicados importantes o encuestas al ingresar al portafolio estudiante, para enterarme inmediatamente de información urgente o encuestas.	
Criterios de Aceptación:	CA1: Al ingresar al portafolio estudiante en la página del calendario me deben aparecer comunicados pendientes.	
	CA2: Si existe un comunicado, se debe mostrar una ventana modal que bloquee la interfaz principal hasta que sea cerrada.	
	CA3: El comunicado puede contener texto plano o una acción (botón) que redirija a una encuesta interna o externa.	

Tabla 7 Historia de Usuario 5: Foro General

Código: HU_05	Título: Foro General	Prioridad: Alta
Descripción:	Como estudiante, quiero participar en un Foro General visible para todos los integrantes del curso, con el fin de discutir temas académicos y socializar sin que estas interacciones sean evaluadas.	
Criterios de Aceptación:	CA1: Debe estar siempre disponible durante el semestre. Permite crear hilos, responder a otros y eliminar comentarios propios.	
	CA2: Debe ser accesible desde el menú más del aula virtual.	
	CA3: La interfaz debe mostrar la anidación de respuestas para seguir el hilo de la conversación.	

Tabla 8 Historia de Usuario 6: Foro Actividad

Código: HU_06	Título: Foro Actividad	Prioridad: Alta
Descripción:	Como estudiante, quiero participar en un Foro de tipo Actividad, con el fin de cumplir con una actividad académica y que estas interacciones sean calificadas.	
Criterios de Aceptación:	CA1: Debe estar disponible para acceder desde el calendario de actividades.	
	CA2: Debe ser accesible, aunque el tiempo de una actividad esta vencido.	
	CA3: Debe permitir ver y eliminar respuestas propias a pesar de que el tiempo de la actividad haya acabado.	

Tabla 9 Historia de Usuario 7: Evaluación Docentes

Código: HU_07	Título: Evaluación del Desempeño Docente	Prioridad: Alta
Descripción:	Como estudiante, quiero evaluar a mis docentes mediante un formulario digital.	

Criterios de Aceptación:	CA1: El sistema debe listar los docentes pendientes de evaluación para el periodo actual.
	CA2: Al ingresar al formulario deben cargar todas las preguntas disponibles.
	CA3: Una vez enviada la evaluación, la evaluación se marca como realizada en la lista de pendientes.
	CA4: Esta evaluación no tiene ningún tipo de calificación, ni temporizador para realizarla.

Tabla 10 Historia de Usuario 8: Becas

Código: HU_08	Título: Gestión de Solicitudes de Becas	Prioridad: Alta
Descripción:	Como estudiante, quiero gestionar todo el ciclo de vida de mi solicitud de beca	
Criterios de Aceptación:	CA1: El estudiante puede generar una solicitud inicial. El sistema debe validar automáticamente si cumple los requisitos académicos para el tipo de beca seleccionado.	
	CA2: Si la solicitud es válida, se habilita una segunda pantalla para subir documentos o requisitos necesarios.	
	CA3: Solo si la beca es aprobada, se habilita una página final para ingresar la información de la cuenta bancaria y subir el certificado bancario.	

Tabla 11 Historia de Usuario 9: Evaluación Online

Código: HU_09	Título: Evaluaciones Online	Prioridad: Alta
Descripción:	Como estudiante, quiero rendir mis exámenes y pruebas desde el dispositivo móvil, para cumplir con mis obligaciones académicas con flexibilidad.	
Criterios de Aceptación:	CA1: Se puede ingresar desde el calendario de actividades.	

CA2: La evaluación debe tener un temporizador visible que cierra el intento automáticamente al llegar a cero.

CA3: Soporte para preguntas de tipo: opción única, opción múltiple, texto y emparejamiento.

CA4: Al finalizar (manual o por tiempo), se debe calificar automáticamente y redirigir a una pantalla de resultados que muestre la nota y el detalle de aciertos/errores.

Cuando los requisitos han sido recopilados, se clasifican y se estiman en el Product Backlog en conjunto con el Product Owner, el Scrum Master y el equipo de desarrollo. La estimación se lleva a cabo mediante una escala que toma como referencia la serie de Fibonacci (1, 2, 3, 5, 8), tal como se describe en "Agile Estimating and Planning" [67]. Los valores finales se muestran en la Tabla 12.

Tabla 12 Product Backlog

Nro. Orden	ID, Historia de Usuario	Descripción	Estimación
1	HU_01	Visualización del Historial de Pagos	2
2	HU_02	Árbol de Contenidos de la Materia	2
3	HU_03	Lista y Descarga de Formatos de Vinculación	3
4	HU_04	Sistema de Avisos Modales	5
5	HU_05	Foro General	8
6	HU_06	Foro Actividad	8
7	HU_07	Evaluación del Desempeño Docente	8

8	HU_08	Gestión de Solicitudes de Becas	8
9	HU_09	Evaluaciones Online	8

2.2 Desarrollo Fase de Juego

El desarrollo se organizó en seis Sprints operativos. La arquitectura tecnológica se mantuvo alineada con la Fase 1, utilizando una arquitectura limpia en Flutter para el cliente móvil y microservicios en NestJS para el backend, garantizando escalabilidad y mantenibilidad del sistema.

Sprint 1: Pagos y Contenidos de la materia

- Planificación - Sprint 1

La reunión que se llevó a cabo con el equipo scrum dio como resultado el sprint backlog que se muestra en la Tabla 13.

Tabla 13 Sprint Backlog – Sprint 1

Historia de Usuario	Tareas Técnicas	Estimación (Horas)
HU_01: Pagos	Implementación del servicio API para consulta de historial financiero.	2
	Lógica para descarga y visualización de PDF (Facturas/Matrículas).	3
	Desarrollar pruebas unitarias en API.	1
	Diseño de la interfaz de lista de pagos (Card widget).	3
	Añadir la posibilidad de realizar scroll infinito en la lista.	2
HU_02: Contenidos	Implementar el endpoint para consulta de contenidos de la materia.	1

	Desarrollar las pruebas unitarias en la API.	1
	Diseño de la interfaz de árbol (TreeView) para unidades y temas.	3
	Manejo de estados de carga y error en la visualización de la estructura.	2
Reuniones	Planificación	1.5
	Revisión	2
	Retrospectiva	1
	Daily scrum	1.5
	Total	24

- Incremento - Sprint 1

En la Figura 8 se muestra el listado de transacciones que el estudiante posee como se solicita en la Tabla 3.

Figura 8 Listado del historial de transacciones del estudiante



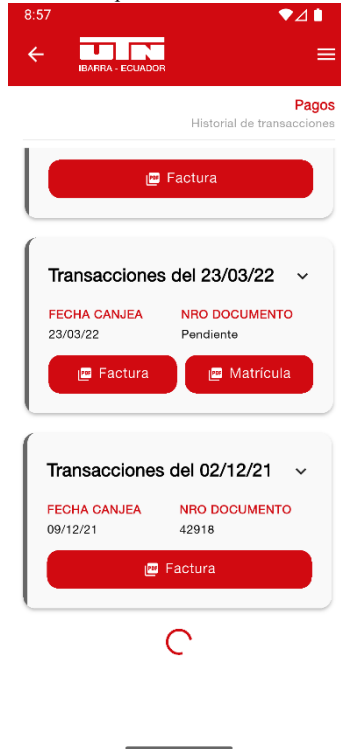
La Figura 9 muestra una transacción cuando esta se despliega mostrando el detalle completo de la transacción tal como se requiere en la Tabla 3.

Figura 9 Detalle expandido de una transacción específica



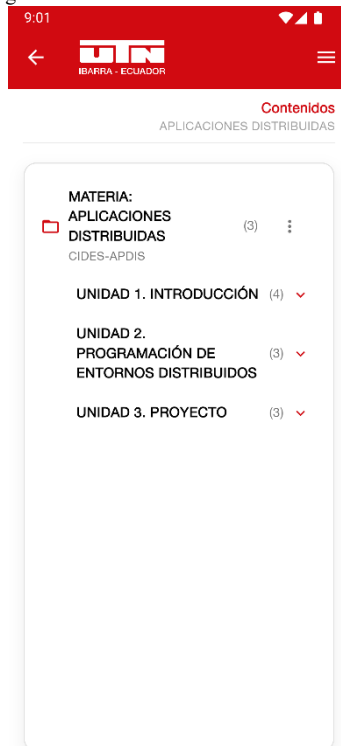
En la Figura 10 se muestra cómo es posible realizar scroll infinito en el listado de transacciones y si existen más transacciones estas cargarán automáticamente.

Figura 10 Scroll infinito para la lista de transacciones disponibles



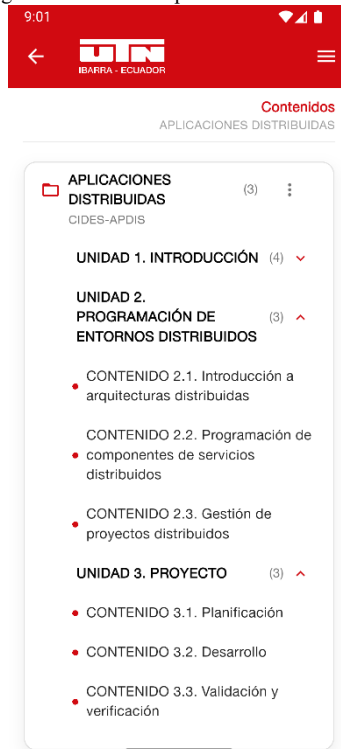
En la Figura 11 se muestra la vista de Contenidos de una materia específica, esta se muestra en forma de un árbol de contenidos tal cual se pidió en la Tabla 4.

Figura 11 Árbol de contenidos de la materia



En la Figura 12 se ve como el contenido una vez ha sido expandido, muestra la planificación curricular de su materia correctamente.

Figura 12 Vista de la planificación curricular



- Revisión - Sprint 1

La Tabla 14 muestra el cumplimiento de los criterios de aceptación de todas las historias de usuario desarrolladas en este Sprint 1.

Tabla 14 Criterios de Aceptación Sprint 1

Historia de Usuario	Criterio de Aceptación	Cumple
H1: Visualización del Historial de Pagos	CA1: El sistema debe listar todas las transacciones históricas del estudiante, ordenadas descendientemente por fecha.	Si
	CA2: Cada ítem de la lista debe mostrar: Concepto, Monto, Fecha y Estado.	Si
	CA3: Si la transacción dispone de archivos asociados (factura electrónica o comprobante de matrícula),	Si

	se debe habilitar un botón para poder acceder a descargar este documento.	
H2: Árbol de Contenidos de la Materia	CA1: Al ingresar a esta sección, se debe mostrar una vista de árbol o lista expandible que represente las Unidades y sus respectivos Temas.	Si
	CA2: La estructura debe reflejar fielmente la planificación curricular.	Si

- Retrospectiva – Sprint 1

En la reunión retrospectiva del sprint se destacó que el incremento funcional presentó correctamente el historial de pagos y permitió la descarga de comprobantes. Por otro lado, se recomendaron ajustes de color para mantener la identidad institucional y una mejor selección de componentes de texto para evitar que estos se confundieran con botones que ejecutan acciones en el sistema.

Sprint 2: Formatos Vinculación y Avisos Modales

- Planificación - Sprint 2

Tabla 15 Sprint Backlog – Sprint 2

Historia de Usuario	Tareas Técnicas	Estimación (Horas)
	Implementar el endpoint en Api para consulta de listado de documentos de vinculación	2.5
HU_03: Formatos Vinculación	Desarrollar las pruebas unitarias en la API	1
	Creación de vista para listado de formatos de vinculación.	2
	Implementación de lógica de descarga directa de documentos o permitir abrir	2.5

	con una aplicación compatible del teléfono.	
HU_04: Avisos Modales	Implementación del endpoint en API para consulta de avisos/comunicados vigentes.	2
	Lógica para marcar un aviso como visto.	1
	Implementación de endpoint para encuestas dinámicas	3
	Desarrollar pruebas unitarias en API	1
	Diseño del componente Modal (Pop-up) con soporte para imágenes y HTML básico.	4
	Diseño de una página capaz de renderizar diferentes tipos de pregunta para lograr una encuesta dinámica	5
Reuniones	Planificación	1.5
	Revisión	2
	Retrospectiva	1
	Daily scrum	1.5
Total		30

- Incremento - Sprint 2

En la Figura 13 se muestra el listado de los formatos de vinculación disponible como se indica en la Tabla 5.

Figura 13 Listado de los formatos de vinculación



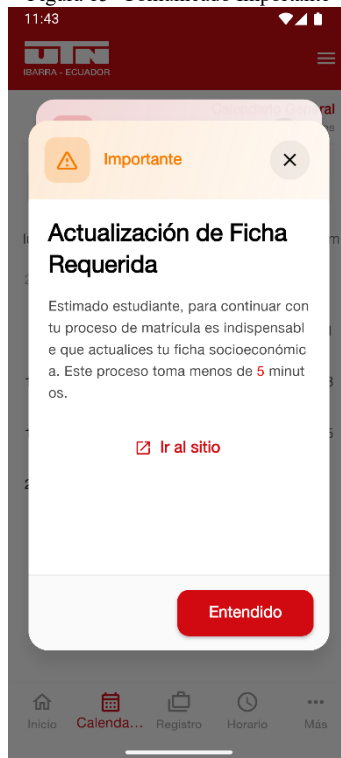
En la Figura 14 se muestra el modal que permite elegir que hacer con el archivo, como se indica en la Tabla 5 ya sea abrir con una aplicación compatible o descargar directamente en el dispositivo, por defecto el documento se descargara en la carpeta de descargas.

Figura 14 Modal de opciones

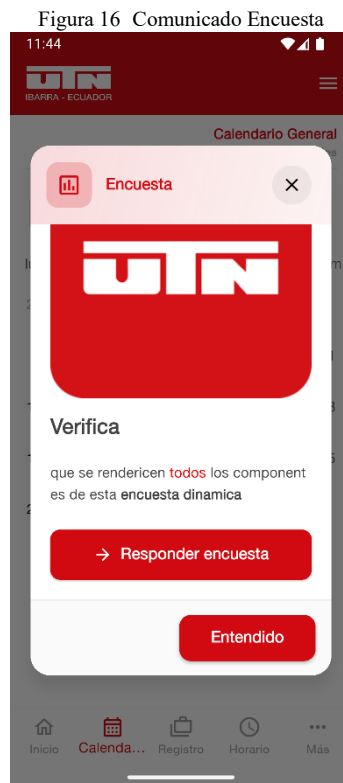


Al ingresar al portafolio estudiante en la página del calendario de actividades, si es que hay comunicados disponibles estos cargarán automáticamente como se observa en la Figura 15 esto se indica en la Tabla 6, si hay varios estos se apilarán como tarjetas.

Figura 15 Comunicado Importante

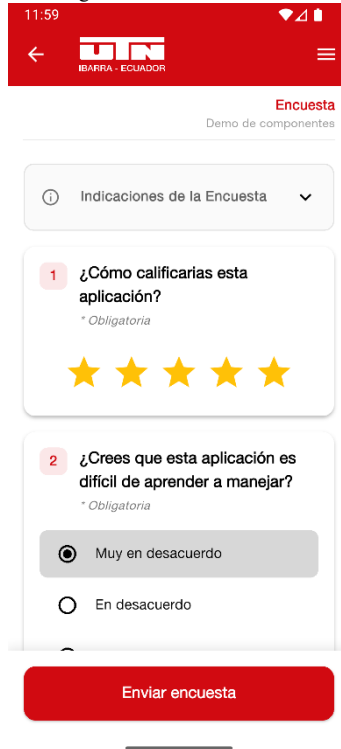


En la Figura 16 se muestra un comunicado encuesta como se indica en la Tabla 6, este tiene un botón de acción que me redirecciona a una encuesta en la aplicación.



En la Figura 17 se muestra cómo se carga las preguntas de la encuesta dentro de la misma aplicación.

Figura 17 Encuesta interna



- Revisión - Sprint 2

A continuación, en la Tabla 16 se detalla el cumplimiento de los criterios de aceptación de todas las historias de usuario desarrolladas durante este Sprint.

Tabla 16 Criterios de Aceptación Sprint 2

Historia de Usuario	Criterio de Aceptación	Cumple
	CA1: El sistema debe mostrar una lista clara de archivos disponibles.	Si
H3: Descarga de Formatos de Vinculación	CA2: Cada ítem de la lista debe mostrar el nombre del archivo y un icono representativo de su tipo (Word o Excel).	Si
	CA3: Al hacer presionar en un archivo, este debe dejar elegir si lo quiere descargar automáticamente al dispositivo, o la opción para	Si

	abrir con una aplicación compatible.	
H4: Sistema de Avisos Modales	CA1: Al ingresar al portafolio estudiante en la página del calendario me deben aparecer comunicados si es que están pendientes.	Si
	CA2: Si existe un comunicado, se debe mostrar una ventana modal que bloquee la interfaz principal hasta que sea cerrada.	Si
	CA3: El comunicado puede contener texto plano o una acción (botón) que redirija a una encuesta interna o externa.	Si

- Retrospectiva – Sprint 2

En la reunión que se realizó se destacó como acierto el haber agregado la posibilidad de que, si existen varios comunicados disponibles en el mismo momento, estos se apilaran como tarjetas, además de que estos tuvieran una interfaz simple pero funcional.

Sprint 3: Foro Online

- Planificación - Sprint 3

Tabla 17 Sprint Backlog – Sprint 3

Historia de Usuario	Tareas Técnicas	Estimación (Horas)
HU_05: Foro General	Implementación de un endpoint en la API.	4
	Lógica para anidar a comentario padre con su respectivo hijo	1
	Lógica para anidar a un usuario con todos sus	1

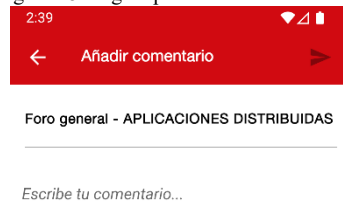
	comentarios para mejorar la estructura del JSON	
	Desarrollar pruebas unitarias en API.	2
	Desarrollar funcionalidad base de comentarios anidados en la APP.	4
	Desarrollar indentación para mejorar el diseño de los comentarios anidados.	4
	Desarrollar página auxiliar para comentarios que superen el nivel 3.	4
	Desarrollar página para crear/editar un comentario con la funcionalidad de texto enriquecido.	5
	Desarrollar un banner de contexto en la página de responder comentario.	2
	Desarrollar la funcionalidad de poder eliminar un comentario propio.	1
HU_06: Foro Actividad	Desarrollo de lógica para diferenciar Foro General vs. Foro Actividad.	2
	Implementación de validación de fechas para habilitar/deshabilitar creación/edición de más comentarios en una actividad.	2
	Reutilización de los widgets creados para Foro General	2
Reuniones	Planificación	1.5

Revisión	2
Retrospectiva	1
Daily scrum	1.5
Total	40

- Incremento - Sprint 3

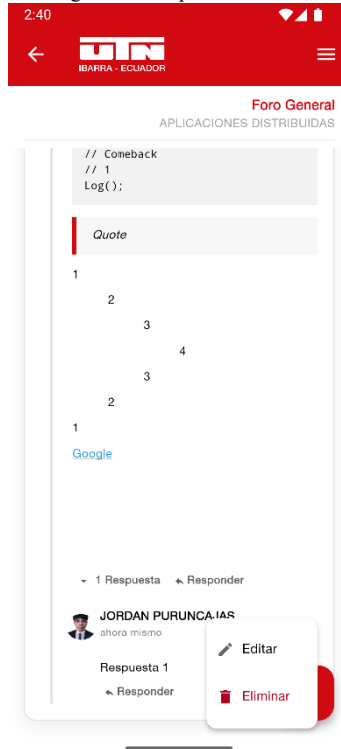
En la Figura 18 se muestra la página para crear un comentario en el Foro General en un aula virtual, este mismo siempre es accesible desde el menú lateral de cada aula virtual como se indica en la Tabla 7.

Figura 18 Página para crear un comentario



En la Figura 19 se muestra como una respuesta a un comentario se anida automáticamente siguiendo la jerarquía de padre e hijo, como se indica en la Tabla 7, además de que se indica en la Figura 19 como aparece un menú de 3 puntos en el comentario que me permite editar o eliminar, este solo aparece al propietario del comentario.

Figura 19 Respuesta anidada



En la Figura 20 se muestra la página para responder a un comentario específico, además que en esta página se incluye opciones de texto enriquecido como se muestra en la Figura 21, esto para darle más estilos a los comentarios.

Figura 20 Página para responder a un comentario específico

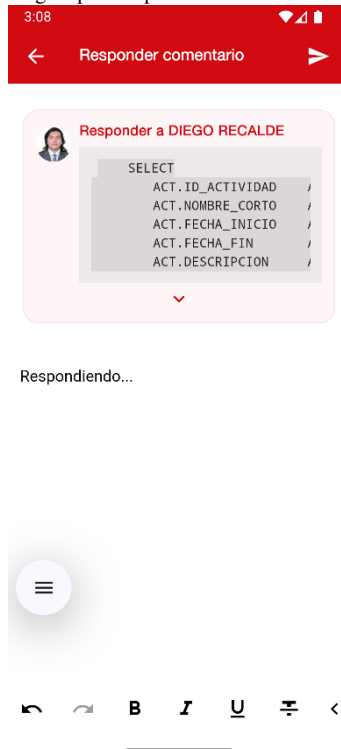
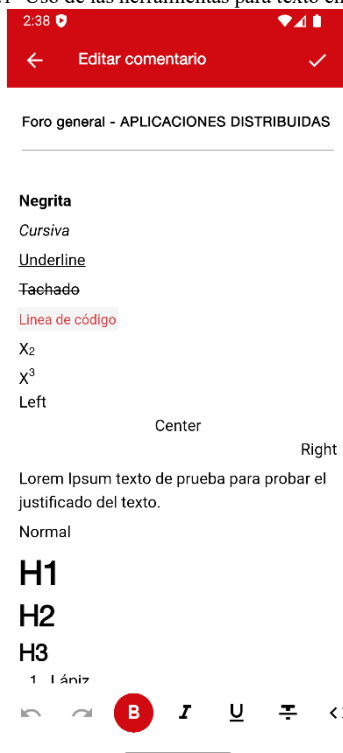
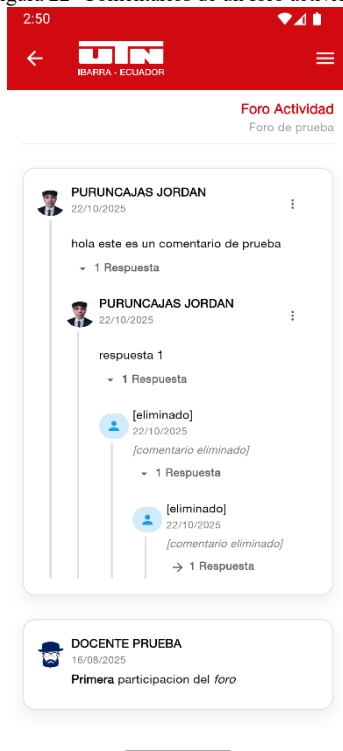


Figura 21 Uso de las herramientas para texto enriquecido

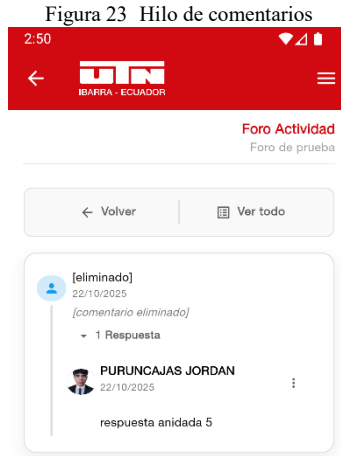


En la Figura 22 se muestra la página de un foro de tipo actividad, este siempre es accesible desde el calendario de actividades a pesar de que el tiempo de la actividad haya vencido tal y como se pide en la Tabla 8.

Figura 22 Comentarios de un foro actividad

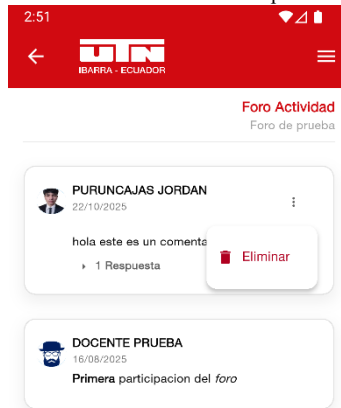


En la Figura 23 se muestra la página de hilo de comentarios esta página está disponible cuando un comentario supera el nivel de anidación 3, entonces los demás comentarios cargarán en una vista aparte para evitar el desbordamiento del contenido.



En la Figura 24 se muestra como en un foro de tipo actividad no se puede editar ni crear un comentario porque el tiempo de actividad ya venció, además que dé al propietario solo le permite eliminar, como se indica en la Tabla 8.

Figura 24 Comentario con menú de opciones limitado



- Revisión - Sprint 3

Tabla 18 Criterios de Aceptación Sprint 3

Historia de Usuario	Criterio de Aceptación	Cumple
H5: Foro General	CA1: Debe estar siempre disponible durante el semestre. Permite crear hilos, responder a otros y eliminar comentarios propios.	Si
	CA2: Debe ser accesible desde el menú más del aula virtual.	Si
	CA3: La interfaz debe mostrar la anidación de respuestas para seguir el hilo de la conversación.	Si
H6: Foro Actividad	CA1: Debe estar disponible para acceder desde el calendario de actividades.	Si
	CA3: Debe permitir ver y eliminar respuestas propias a	Si

pesar de que el tiempo de la actividad haya acabado.

- Retrospectiva – Sprint 3

Se resaltó como un acierto el que haya considerado hacer que los comentarios que superen un nivel de anidación 3 se vayan a otra vista para evitar que el contenido se desborde de la pantalla o se vea muy agrupado. Esto permitió tener una mejor experiencia para el usuario, alineándose con las buenas prácticas de diseño.

Sprint 4: Evaluación Docentes

- Planificación - Sprint 4

Tabla 19 Sprint Backlog – Sprint 4

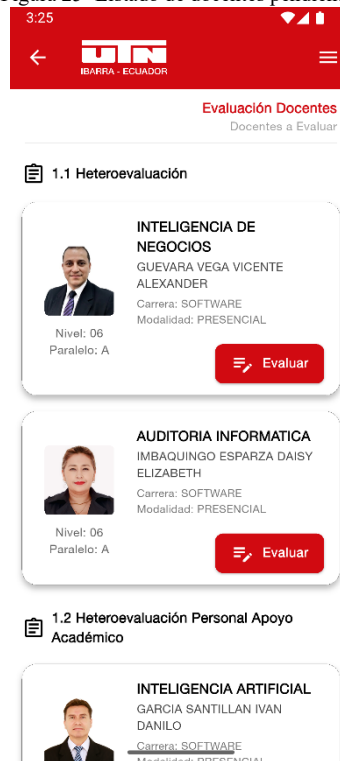
Historia de Usuario	Tareas Técnicas	Estimación (Horas)
	Implementación de un endpoint en la API para conocer que docentes se evaluarán.	3
	Implementación de un endpoint en la API para traer el formulario completo.	2.5
	Implementación de un endpoint en la API para traer las alternativas de un formulario si es que lo tiene.	3.5
HU_07: Evaluación Docente	Desarrollar pruebas unitarias en API.	2.5
	Desarrollar la funcionalidad de listar los docentes disponibles a evaluar	2
	Desarrollar los diferentes tipos de preguntas para las evaluaciones, opción única/múltiple y abierta	3.5
	Desarrollar contenedores desplegados que indiquen	2

	más información acerca la evaluación	
	Desarrollar un dialog de confirmación para finalizar la evaluación	1
Reuniones	Planificación	1.5
	Revisión	2
	Retrospectiva	1
	Daily scrum	1.5
Total		26

- Incremento - Sprint 4

En la Figura 25 se muestra la lista de docentes pendientes a ser evaluados en el ciclo actual, tal cual como se indica en la Tabla 9.

Figura 25 Listado de docentes pendientes



En la Figura 26 y Figura 27 se indica un cajón desplegable que me muestra más información acerca del formulario de evaluación.

Figura 26 Cajón desplegable de las indicaciones de evaluación

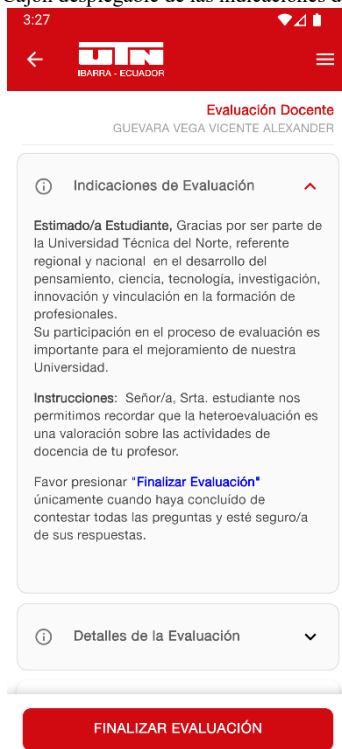
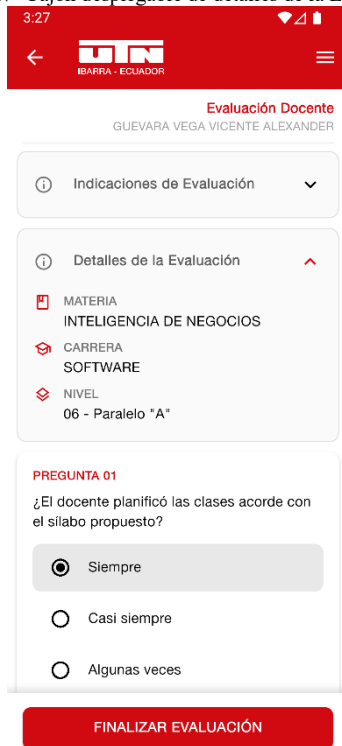


Figura 27 Cajón desplegable de detalles de la Evaluación



En la Figura 28 se muestra como cargan las preguntas correctamente para el formulario de cada docente, como se indica en la Tabla 9.

Figura 28 Vista de las preguntas del formulario

2:14

← UIN IBARRA - ECUADOR ≡

Evaluación Docente
GARCIA SANTILLAN IVAN DANILO

AMBITO: PLANEACION ACADÉMICA.
PREGUNTA 01
¿El docente planificó las clases acorde con el sílabo propuesto?

Siempre

Casi siempre

Algunas veces

Rara vez

Nunca

COMPONENTE: DIDÁCTICA
PREGUNTA 02
¿El docente realizó las actividades didácticas de la asignatura o módulo de forma coherente con los contenidos del sílabo?

FINALIZAR EVALUACIÓN

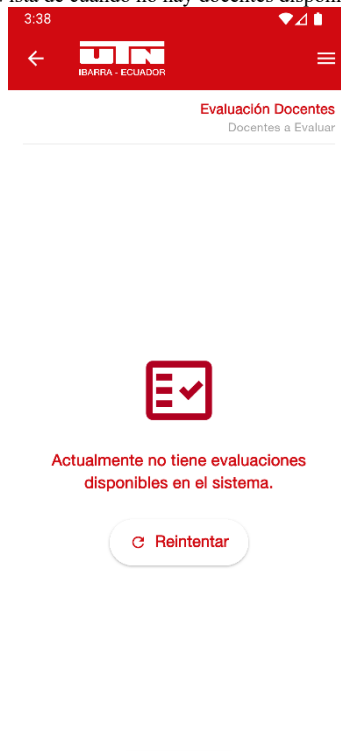
En la Figura 29 se muestra como una vez un docente fue evaluado este es marcado como “Evaluado”, para así saber cuáles son los docentes que quedan aún pendientes por responder, como se pide en la Tabla 9.

Figura 29 Vista de docentes marcados como evaluados



En la Figura 30 se observa que una vez hayan sido evaluados todos los docentes, la vista muestra un mensaje de que ya no hay más docentes a evaluar.

Figura 30 Vista de cuando no hay docentes disponibles a evaluar



- Revisión - Sprint 4

En la Tabla 20 se muestra la revisión de los criterios de aceptación para este Sprint.

Tabla 20 Criterios de Aceptación Sprint 4

Historia de Usuario	Criterio de Aceptación	Cumple
H7: Evaluación del Desempeño Docente	CA1: El sistema debe listar los docentes pendientes de evaluación para el periodo actual.	Si
	CA2: Al ingresar al formulario deben cargar todas las preguntas disponibles.	Si
	CA3: Una vez enviada la evaluación, la evaluación se marca como realizada en la lista de pendientes.	Si
	CA4: Esta evaluación no tiene ningún tipo de calificación, ni temporizador para realizarla.	Si

- Retrospectiva – Sprint 4

En la reunión que se llevó a cabo se destacó el uso de cajones desplegados para mostrar más información acerca de la evaluación al docente que se está realizando, ya que al permitir desplegar y contraer este contenido se evita mostrar mucha información cuando no es necesario. Además, se confirmó que este proceso de evaluar al docente cumple con el proceso ya establecido para la evaluación en el portafolio web.

Sprint 5: Becas

- Planificación - Sprint 5

Tabla 21 Sprint Backlog – Sprint 5

Historia de Usuario	Tareas Técnicas	Estimación (Horas)
HU_08: Becas	Implementación de endpoint en la API para el proceso activo de becas.	2
	Implementación del endpoint para validar el tipo de beca.	1

	Implementación del endpoint en la API para subir requisitos y datos de la cuenta bancaria.	3
	Desarrollar pruebas unitarias en API.	2
	Desarrollar de un menú para el proceso de becas.	2
	Desarrollar de la vista para solicitar la beca.	1
	Desarrollar un selector para el tipo de beca que valide con el backend.	2
	Desarrollar vista para ver la información de la solicitud de beca generada.	1
	Desarrollar vista para ingresar los requisitos dependiendo el tipo de beca.	2
	Desarrollar vista para ver la información del contrato de beca.	2
	Desarrollar vista para ingresar los datos de la cuenta bancaria.	2
Reuniones	Planificación	1.5
	Revisión	2
	Retrospectiva	1
	Planificación	1.5
Total		26

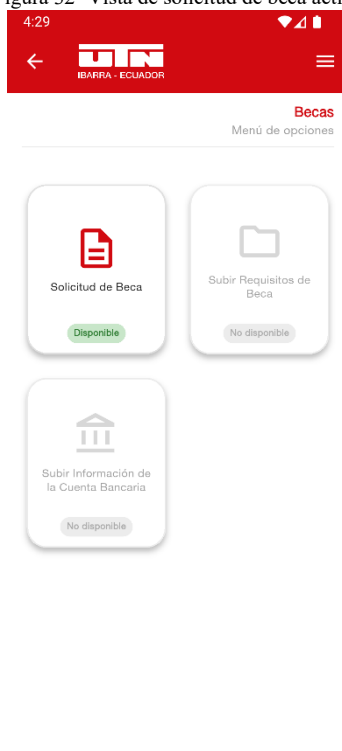
- Incremento - Sprint 5

En la Figura 31 muestra el menú de becas con todas sus opciones inhabilitadas porque no hay una convocatoria de beca activa, mientras que en la Figura 32 muestra que la opción “Solicitud de Beca” se activa cuando hay una convocatoria de beca activa.

Figura 31 Vista del menú de opciones de becas

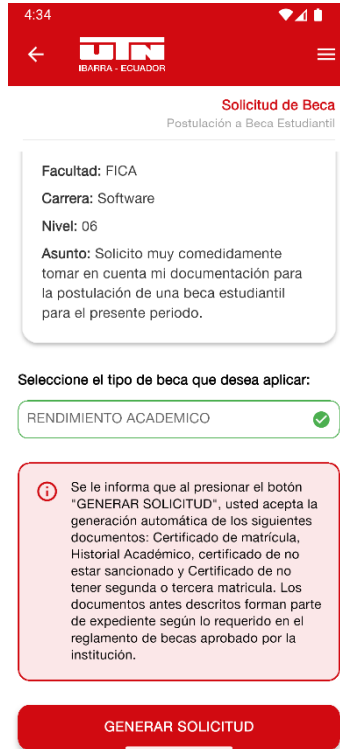


Figura 32 Vista de solicitud de beca activa



En la Figura 33 muestra la vista de solicitud de beca, donde el estudiante puede postular a distintos tipos becas, estos tipos serán validados automáticamente cuando sean seleccionados.

Figura 33 Vista del formulario de solicitud de beca



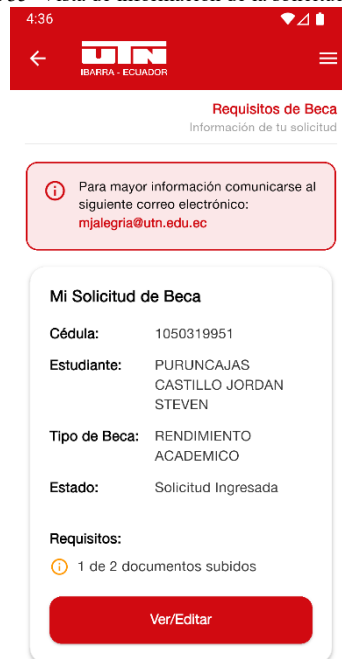
En la Figura 34 se muestra la opción “Subir Requisitos de Beca” habilitada, una vez que se haya generado exitosamente la solicitud de beca.

Figura 34 Vista de subir requisitos activa



En la Figura 35 se muestra la información de la solicitud de beca previamente generada en forma de resumen.

Figura 35 Vista de información de la solicitud de beca



En la Figura 36 se muestra el formulario para subir los documentos requeridos, estos campos pueden variar dependiendo el tipo de beca, para guardar los cambios se debe aceptar obligatoriamente los términos y condiciones.

Figura 36 Formulario de requisitos requeridos



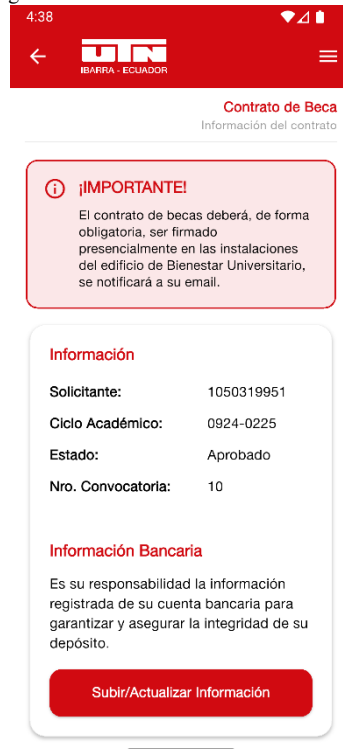
En la Figura 37 muestra la opción “Subir Información de la Cuenta Bancaria Activa”, esta opción se habilita únicamente cuando la beca ha sido aprobada.

Figura 37 Vista subir información bancaria activa



En la Figura 38 muestra información sobre el contrato de la beca a manera de resumen, permitiendo acceder a subir o actualizar los datos bancarios.

Figura 38 Vista de información del contrato



En la Figura 39 muestra los campos obligatorios que se necesita para poder guardar la información bancaria del estudiante.

Figura 39 Formulario de información bancaria

4:39

IBARRA ECUADOR

Cuenta Bancaria
Completa tus datos bancarios

Entidad Bancaria
BANCO DEL PICHINCHA

Nro. Cuenta
111111111

Tipo Cuenta
Ahorros

Certificado Bancario / Libreta (PDF)
Adjunte el documento PDF que certifique sus datos.

test-grises.pdf

Cambiar archivo (PDF)

GUARDAR

- Revisión - Sprint 5

Tabla 22 Criterios de Aceptación Sprint 5

Historia de Usuario	Criterio de Aceptación	Cumple
H8: Gestión de Solicitudes de Becas	CA1: El estudiante puede generar una solicitud inicial. El sistema debe validar automáticamente si cumple los requisitos académicos para el tipo de beca seleccionado.	Si
	CA2: Si la solicitud es válida, se habilita una segunda pantalla para subir documentos o requisitos necesarios.	Si
	CA3: Solo si la beca es aprobada, se habilita una página final para ingresar la información de la cuenta bancaria y subir el certificado bancario.	Si

- Retrospectiva – Sprint 5

Esta reunión se hizo en colaboración con un miembro del DDTI encargado del módulo de becas para el portafolio web, sus observaciones fueron de gran ayuda y verificó que lo desarrollado cumpliera con lo ya establecido en el portafolio web. Además, se destacó que la adaptación de la interfaz fue muy acertada, ya que mantiene coherencia con el resto de la aplicación.

Sprint 6: Evaluación Online

- Planificación - Sprint 6

Tabla 23 Sprint Backlog – Sprint 6

Historia de Usuario	Tareas Técnicas	Estimación (Horas)
HU_09: Evaluación Online	Implementación de un endpoint en la API para el formulario de evaluación.	5
	Implementación de un endpoint para finalizar la evaluación.	2
	Implementación de un endpoint para obtener el resultado de nota con detalle de preguntas y respuestas.	4
	Desarrollar pruebas unitarias en la API.	2.5
	Desarrollar los diferentes tipos de preguntas para las evaluaciones, opción única/múltiple, abierta y emparejamiento.	4
	Desarrollar un temporizador arrestable y desplegable.	1
	Desarrollar mecanismos de seguridad (bloqueo de retroceso, autoenvío al finalizar tiempo).	1

	Implementar mecanismo que detecte en tiempo real actividad sospechosa dentro de la evaluación.	3.5
	Implementar mecanismo que evite captura de imagen/video dentro de la evaluación y vista de resultado.	2.5
	Reutilizar un dialog de confirmación para finalizar la evaluación.	0.5
Reuniones	Planificación	1.5
	Revisión	2
	Retrospectiva	1
	Daily scrum	1.5
	Total	32

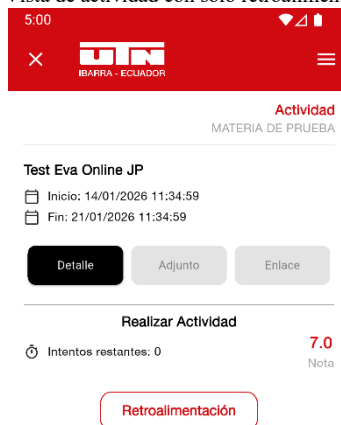
- Incremento - Sprint 6

En la Figura 40 se muestra cómo el contenido de una actividad cambia según su tipo. En el caso de las evaluaciones aparecen dos opciones nuevas: iniciar evaluación y retroalimentación, cada una con una función distinta que se explicará más adelante. En la Figura 41 se muestra que, una vez vencido el tiempo de entrega, solo aparece el botón de “Retroalimentación”, siempre y cuando la evaluación haya sido realizada; de lo contrario, no se muestra ninguna opción.

Figura 40 Vista de actividad para evaluación



Figura 41 Vista de actividad con solo retroalimentación activa



En la Figura 42 se muestra el formulario de evaluación que se puede ingresar desde la vista anterior, en este formulario se muestran diferentes tipos de componentes uno de los importantes es el temporizador este empieza su cuenta regresiva una vez el

formulario de evaluación haya cargado, otro componente es el tipo de pregunta opción única que solo acepta una respuesta y el último es el componente de pregunta textual que permite ingresar contenido libre.

Figura 42 Formulario con tipo de pregunta opción única y textual

The screenshot shows a mobile application interface for an online evaluation. At the top, there is a red header with the logo of UTEC (Universidad Tecnológica de Ecuador) and the text 'IBARRA - ECUADOR'. Below the header, there is a green bar with a clock icon and the text 'Tiempo: 00:58:27' and 'Evaluación en Línea' with a subtext 'Evaluación Online Jordan'. The main content area contains three questions:

- Pregunta 1:** 'Raíz cuadrada de 4'. It has four radio button options: 2 (selected), 1, 20, and 0.
- Pregunta 2:** '¿Qué significa SPA?'. It has a text input field containing 'Single page application.'.
- Pregunta 3:** '¿Qué es SPA?'. It is partially visible and has a text input field.

At the bottom of the screen, there is a red button with the text 'FINALIZAR EVALUACIÓN'.

En la Figura 43 se muestra el componente de pregunta emparejamiento de columnas, este tipo de preguntas permite seleccionar de una lista de opciones previamente ya cargadas por el docente.

Figura 43 Formulario con tipo de pregunta emparejamiento de columnas

5:15

UN IBARRA - ECUADOR

Tiempo: 00:55:51 Evaluación en Línea
Evaluación Online Jordan

Pregunta 4
Capitales de provincias

Imbabura
Ibarra

Carchi
Tulcan

Azuay
Cuenca

Pichincha
Quito

Guayas
Guayaquil

El Oro
Machala

FINALIZAR EVALUACIÓN

En la Figura 44 se muestra el componente de pregunta de opción múltiple, ese tipo de preguntas permite seleccionar una o varias respuestas al mismo tiempo.

Figura 44 Formulario con tipo de pregunta opción múltiple

5:19

UN IBARRA - ECUADOR

Tiempo: 00:51:11 Evaluación en Línea
Evaluación Online Jordan

Pregunta 8
¿Que significa CSS?

Computer Style Syntax

Hojas de Estilo en Cascada

Creative Styling System

Cascading Style Sheets

Pregunta 9
Cuantos huesos tiene el ser humano?

206

208

210

FINALIZAR EVALUACIÓN

En la Figura 45 se muestra un modal de confirmación este aparece al momento de presionar el botón “Finalizar Evaluación”, para asegurar si el estudiante está seguro o no

de ya terminar su intento, una vez aceptado no se puede cancelar y la evaluación se calificará automáticamente

Figura 45 Modal de finalización de evaluación



En la Figura 46 se muestra la nota obtenida al momento de finalizar le evaluación y también muestra un detalle por pregunta en forma de lista para saber su puntuación individual por pregunta.

Figura 46 Vista del resultado de evaluación



- Revisión - Sprint 6

Tabla 24 Criterios de Aceptación

Historia de Usuario	Criterio de Aceptación	Cumple
H9: Evaluaciones Online	CA1: Se puede ingresar desde el calendario de actividades.	Si
	CA2: La evaluación debe tener un temporizador visible que cierra el intento automáticamente al llegar a cero.	Si
	CA3: Soporte para opción única, opción múltiple, texto y emparejamiento.	Si
	CA4: Al finalizar (manual o por tiempo), se debe calificar automáticamente y redirigir a una pantalla de resultados que	Si

muestre la nota y el detalle de aciertos/errores.

- Retrospectiva – Sprint 6

En la reunión realizada se destacó la inclusión de varias funcionalidades para evitar el plagio, como bloquear la captura o grabación de pantalla mientras se está en el formulario de evaluación y la detección en tiempo real de actividades sospechosas, como minimizar la aplicación, usar el panel de notificaciones, el centro de control o cualquier aplicación en modo ventana, para evitar detección excesivamente agresiva, se implementó un período de espera de 3 segundos. Por otro lado, se comprobó que todas las funcionalidades incluidas en el proyecto fueron implementadas correctamente, logrando las metas propuestas y marcando un progreso importante en el desarrollo.

2.3 Desarrollo Fase de Post-Juego

En esta fase final se completó el despliegue del prototipo funcional que incluye las nuevas características de la Fase 2. A diferencia de la fase 1 del proyecto, en este caso se usaron los recursos de infraestructura que ya estaban listos, aprovechando la configuración de contenedores de Docker para tener un entorno de ejecución estable y fácil de manejar. Esto ayudó a ahorrar tiempo al reutilizar lo que ya funcionaba bien.

El despliegue se hizo usando los archivos Dockerfile y docker-compose.yml que se heredaron y se ajustaron para esta versión. Este proceso permitió correr el prototipo en un entorno de desarrollo controlado, siguiendo los mismos estándares de la fase anterior. El objetivo principal era revisar que los módulos nuevos funcionaran bien junto con los microservicios que ya estaban en producción, asegurando que todo el sistema trabajara de forma coordinada.

Este mismo entorno se usó después para hacer las pruebas funcionales y de seguridad que se explican más adelante. Estas pruebas son muy importantes porque ayudan a confirmar que funciones como la gestión de comentarios en foros o la realización de evaluaciones en línea cumplan con lo que se esperaba, según lo que se definió en el backlog. Además, estas pruebas sirvieron para encontrar y arreglar problemas antes de poner todo en funcionamiento definitivo.

III. VALIDACIÓN DE RESULTADOS

3.1 Definición de Modelo de Calidad en Uso

Junto con el Product Owner y el Scrum Master, se definieron las características de calidad en uso que se usarían según la Norma ISO/IEC 25010 de la familia ISO/IEC 25000 [58]. Las características elegidas fueron: Eficacia, Eficiencia y Satisfacción. También se definieron subcaracterísticas y se les dio un peso según su importancia para el negocio. Esto se hizo tomando en cuenta la experiencia que ya tenía el Product Owner en proyectos parecidos.

Tabla 25 Modelo de calidad en uso

Modelo de Calidad en Uso			
Característica	Sub Característica	Peso de característica	Peso de Sub Característica
Eficacia	Tareas completas	42%	16%
	Objetivos logrados		16%
	Tareas sin errores		10%
Eficiencia	Tiempo de tareas	32%	16%
	Eficiencia del tiempo		16%
Satisfacción	Utilidad	26%	10%
	Confianza		8%
	Comodidad		8%

3.2 Medición de Calidad en Uso

Para medir las características y subcaracterísticas se usó la norma ISO/IEC 25022 [61]. La eficacia y eficiencia se evaluaron con un taller práctico, y la satisfacción se midió usando la encuesta SUS.

3.2.1 Definición de Muestra poblacional

La muestra se formó usando un muestreo no probabilístico por conveniencia debido a la gran cantidad de estudiantes que hay en la Universidad. Se eligieron 70 estudiantes de la carrera de Software de la Facultad de Ciencias Aplicadas (FICA), como aparece en el **Anexo A**. En este grupo se encontraron 6 usuarios expertos que ya tienen experiencia usando el portal web de la institución y conocen bien la aplicación. Estos estudiantes son los que tienen los id 13, 17, 24, 30, 43 y 57 como se puede ver en el **Anexo C** y **Anexo D**.

3.2.2 Taller Práctico

El taller práctico se diseñó con 7 objetivos, donde cada objetivo tiene una tarea general que se divide en 3 subtareas, como aparece en la Tabla 26. Para hacer el taller, el usuario debe iniciar sesión en la aplicación y luego ingresar al portafolio estudiante. Todos los objetivos empezaron desde la pantalla principal (Calendario de actividades).

Tabla 26 Taller práctico

Nro.	Objetivo	Tarea
1	Ver comunicados	Al acceder al Portafolio Estudiante gestionar Comunicados : 1. Visualizar enlace del comunicado importante . 2. Visualizar el comunicado y contestar la encuesta obligatoria . 3. Visualizar contenido del comunicado informativo .
2	Ver pagos	En el menú principal (parte inferior), seleccionar la opción Más y luego Pagos : 1. Deslizar hacia abajo para visualizar la lista de todas sus transacciones. 2. Seleccionar/expandir un pago para ver su detalle. 3. Seleccionar la opción Factura o Matrícula para visualizar el documento.
3	Ver formatos de vinculación	En el menú principal (parte inferior), seleccionar la opción Más y luego Formatos Vinculación : 1. Deslizar hacia abajo para visualizar la lista de todos los formatos. 2. Seleccionar opción " Ver " de cualquier formato del listado. 3. En el modal, elegir Abrir con (visualizar en una aplicación compatible) o Descargar (guardar en la carpeta de descargas).

		En el menú principal, seleccionar la opción Registro Académico y luego Aula Virtual de una materia:
4	Ver contenidos de la materia	<ol style="list-style-type: none"> 1. Seleccionar en el apartado Más (la última pestaña de la barra superior) y luego Contenidos. 2. Presionar sobre los títulos de las unidades para desplegar u ocultar los temas. 3. Usar el menú de tres puntos (⋮) para expandir o contraer todos los contenidos de una sola vez.
		En el menú principal, seleccionar la opción Registro Académico y luego Aula Virtual de una materia:
5	Comentar en Foro General	<ol style="list-style-type: none"> 1. Seleccionar en el apartado Más y luego Foro General. 2. Seleccionar (+) botón de crear (esquina inferior derecha). 3. Escribir un comentario y luego Enviar comentario.
		Buscar y seleccionar la actividad de tipo evaluación en el Calendario de Actividades :
6	Completar una evaluación online	<ol style="list-style-type: none"> 1. Presionar el botón Iniciar Evaluación, leer y aceptar el aviso con las indicaciones para rendir la evaluación. 2. Responder las preguntas de opción única/múltiple, emparejamiento y respuesta abierta que estén disponibles. 3. Finalizar el intento, enviar la evaluación y verificar la nota obtenida.
		Desde el menú principal, seleccionar la opción Más y luego Evaluación Docentes :
7	Evaluar el desempeño de su docente	<ol style="list-style-type: none"> 1. Escoger un docente que esté disponible para ser evaluado. 2. Llenar el formulario evaluación respondiendo todas las preguntas. 3. Presionar en Finalizar Evaluación, confirmar el envío del formulario y verificar el mensaje de confirmación exitosa.

3.2.3 Encuesta SUS

La encuesta System Usability Scale (SUS) incluye 10 preguntas para medir qué tan usable es un sistema, como se muestra en la Tabla 27. Se usa la escala de Likert con 5 opciones, donde 1 significa "Totalmente en desacuerdo" y 5 significa "Totalmente de acuerdo" [68], [69].

Tabla 27 Preguntas Encuestas SUS

Código	Pregunta
P1	¿Considera usted que usaría este sistema frecuentemente?
P2	¿Considera usted que este sistema presenta una complejidad innecesaria?
P3	¿Considera usted que el sistema fue fácil de usar?
P4	¿Considera usted que requeriría el apoyo de una persona con conocimientos técnicos para usar este sistema?
P5	¿Considera usted que las distintas funciones de este sistema están bien integradas?
P6	¿Considera usted que el sistema es demasiado inconsistente?
P7	¿Considera usted que la mayoría de la gente aprendería a usar este sistema muy rápidamente?
P8	¿Considera usted que el sistema es muy engorroso de utilizar?
P9	¿Se siente confiado al usar el sistema?
P10	¿Considera usted que necesita aprender muchas cosas tecnológicas antes de usar el sistema?

Las 10 preguntas del cuestionario SUS se clasifican en dos dimensiones que evalúan características complementarias del sistema:

- Utilidad (P1, P6 y P9)

Este conjunto de preguntas analiza el valor práctico y la efectividad del sistema para el usuario. La pregunta P1 evalúa la disposición de uso frecuente del sistema, P6 verifica la consistencia de su funcionamiento, y P9 mide el grado de confianza que experimenta el usuario al interactuar con la aplicación.

- Comodidad (P2, P3, P4, P5, P7, P8 y P10)

Este grupo de preguntas examina la facilidad de uso y la calidad de la experiencia del usuario. Contempla aspectos como la complejidad percibida (P2), la facilidad de aprendizaje (P3, P7), la necesidad de apoyo técnico (P4), la integración entre funcionalidades (P5), las dificultades de operación (P8) y los requisitos de conocimiento previo (P10).

3.3 Validaciones estadísticas de los instrumentos de medición

3.3.1 Fiabilidad del taller práctico

Para comprobar la fiabilidad del taller práctico, se usó la prueba de Kolmogorov-Smirnov con los datos del Anexo C. Esta prueba ayuda a saber si los datos tienen una distribución normal; si el valor es mayor a 0.5, significa que la distribución es normal.

Figura 47 Test de normalidad taller práctico

	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
Errores	,511	70	,000	,402	70	,000
TareasLogradas	,511	70	,000	,402	70	,000
ObjetivosLogares	,511	70	,000	,400	70	,000

a. Corrección de significación de Lilliefors

3.3.2 Fiabilidad de la encuesta SUS

Después de registrar los datos del Anexo D, se midió la fiabilidad de la encuesta SUS (10 preguntas) con el coeficiente alfa de Cronbach. Según los criterios técnicos, valores entre 0,70 y 0,90 muestran una buena consistencia interna. Por otra parte, resultados mayores a 0,90 pueden significar que hay preguntas repetidas o muy parecidas, y valores menores a 0,70 indican baja consistencia [70]. Por último, se recodificaron las preguntas pares porque están formuladas de manera opuesta a las impares, para que todas tengan el mismo sentido en la escala.

Figura 48 Alfa de Cronbach encuesta SUS

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,858	10

Como se observa en la Figura 48, el coeficiente obtenido es de **0,858**. Este resultado indica una consistencia interna muy alta, situada dentro del rango recomendado de 0,80 a 0,90. Por lo tanto, se considera que la escala es fiable y adecuada para continuar con el análisis de los resultados.

3.4 Evaluación del Modelo de Calidad en Uso

3.4.1 Eficacia

Esta característica mide la precisión y completitud con la que los estudiantes logran los objetivos.

- **Sub característica: Tareas Completadas**

Para calcular esta métrica se hace uso de la siguiente fórmula:

$$X = \frac{A}{B}$$

Donde:

- **A** = Número de subtareas completadas con éxito.
- **B** = Número total de subtareas intentadas.

Reemplazando los valores **A (1437)** y **B (1470)** se obtiene como resultado **0.978**, lo que significa que el **97.8%** de las tareas fueron completadas con éxito.

- **Sub característica: Objetivos Logrados**

Para calcular esta métrica se hace uso de la siguiente fórmula:

$$X = \frac{A}{B}$$

Donde:

- **A** = Número de objetivos cumplidos.
- **B** = Número total de objetivos planteados.

Se establecieron 7 objetivos generales. Siguiendo el criterio de la norma, si una subtarea no se completa, el objetivo que la engloba se considera no alcanzado. Reemplazando los valores **A (478)** y **B (490)** se obtiene como resultado **0.976**, lo que significa que el **97.6%** de los objetivos fueron alcanzados.

- **Sub característica: Errores en una tarea**

Para calcular esta métrica se hace uso de la siguiente fórmula:

$$X = 1 - \frac{A}{B}$$

Donde:

- **A** = Número de errores cometidos por los usuarios.
- **B** = Número total de tareas.

Reemplazando los valores **A (11)** y **B (490)** se obtiene como resultado **0.978**, lo que significa que el **97.8%** de las tareas se realizaron sin errores por parte del estudiante.

3.4.2 Eficiencia

Esta característica mide el esfuerzo que los estudiantes realizaron para completar las tareas del taller en relación con el rendimiento de los expertos.

- **Sub característica: Tiempo de tareas**

$$X = \frac{A}{B}$$

Donde:

- **A:** Tiempo promedio de los usuarios expertos.
- **B:** Tiempo promedio de los usuarios nuevo.

Primero es necesario calcular el tiempo promedio por tarea para los dos grupos de usuarios; las tareas no completadas no se toman en cuenta. Posteriormente, se aplica la fórmula por cada tarea y se calcula el promedio de estos resultados. Finalmente, se obtiene como resultado **(0.610)** lo que significa que existe una **eficiencia del 61%** en los tiempos de tareas.

- **Sub característica: Eficiencia del Tiempo**

Para calcular esta métrica se hace uso de la siguiente fórmula:

$$X = \frac{A}{B}$$

Donde:

- **A** = tiempo empleado por usuarios expertos para completar un objetivo
- **B** = tiempo empleado por usuarios nuevos al completar un objetivo

Primero es necesario calcular el tiempo que tarda cada usuario en completar un objetivo, los objetivos no cumplidos no se toman en cuenta. Posteriormente se calcula el promedio del tiempo por cada objetivo de los dos grupos de usuarios. Se aplica la fórmula

a cada objetivo y se calcula el promedio. Finalmente, se obtiene como resultado **(0.615)** lo que significa que existe una **eficiencia del 62%** en los tiempos de los objetivos.

3.4.3 Satisfacción

Esta característica evalúa la respuesta afectiva y la actitud del estudiante al interactuar con las nuevas funcionalidades implementadas en la Fase 2.

- **Sub característica: Utilidad**

Para medir la utilidad percibida por los estudiantes, se aplicó un peso específico a cada respuesta de la escala de Likert como se muestra en la Tabla 28.

Tabla 28 Peso respuesta encuesta SUS

Escala	Respuesta	Peso
1	Totalmente en desacuerdo	0.2
2	En desacuerdo	0.4
3	Neutro	0.6
4	De acuerdo	0.8
5	Totalmente de acuerdo	1

Es importante destacar que, para el análisis de la pregunta **P6** (al ser de sentido negativo), se invirtió la escala para mantener la coherencia con los indicadores de calidad.

Tabla 29 Resultado SUS - Utilidad

Pregunta	1	2	3	4	5	Resultado
P1	0	1	7	34	28	59.8 de 70
P6	3	3	16	20	28	55.4 de 70
P9	0	1	9	31	29	59.6 de 70
Promedio						58.27 de 70

Para calcular esta métrica se aplicó la siguiente fórmula:

$$X = \frac{A}{B}$$

Donde:

- **A** = Promedio de usuarios que perciben útil el aplicativo.
- **B** = Total de usuarios encuestados.

Reemplazando los valores **A (58.27)** y **B (70)**, se obtiene un resultado de **0.832**, lo que indica que el **83.2%** de los estudiantes perciben que las nuevas funcionalidades del aplicativo son útiles.

- **Sub característica: Confianza**

La confianza mide el grado en que el sistema opera según las expectativas del usuario sin generar incidentes. Se calculó utilizando la ausencia de reclamos durante la ejecución del taller práctico.

Para el cálculo se empleó la fórmula:

$$X = 1 - \frac{A}{B}$$

Donde:

- **A** = Usuarios que presentaron algún reclamo o dificultad crítica (21 usuarios).
- **B** = Total de usuarios evaluados (70 usuarios).

Reemplazando los valores, se obtiene un resultado de **0.70**, lo que representa un nivel de confianza del **70%**.

- **Sub característica: Comodidad**

Se utilizaron los pesos de la Tabla 28 y se invirtió el sentido de las preguntas negativas (**P2, P4, P8, P10**) para normalizar los resultados hacia una escala de satisfacción positiva.

Tabla 30 Resultados SUS - Comodidad

Pregunta	1	2	3	4	5	Resultado
P2	4	11	15	26	14	49.0 de 70
P3	0	1	7	29	33	60.8 de 70
P4	4	11	8	15	32	54.0 de 70
P5	0	1	7	41	21	58.4 de 70
P7	0	1	8	29	32	60.4 de 70
P8	4	7	13	22	24	53.0 de 70
P10	3	6	11	15	35	56.6 de 70
Promedio						56.03 de 70

Se aplicó la fórmula de medición:

$$X = \frac{A}{B}$$

Donde:

- **A** = Usuarios que perciben el aplicativo como cómodo.
- **B** = Total de usuarios encuestados.

Reemplazando los valores **A (56.03)** y **B (70)**, se obtiene un resultado de **0.800**, determinando que el **80.0%** de los usuarios consideran que el aplicativo es cómodo y fácil de utilizar

3.5 Resultados del Modelo de Calidad en Uso

Una vez ya realizado los cálculos de las métricas de todas las subcaracterísticas definidas en la Tabla 25, se pudo validar con éxito la calidad en uso de la Fase 2 del portafolio estudiante. Como se aprecia en la Tabla 31, el resultado obtenido es de **80.98%**.

Como se observa en la Figura 49, el puntaje de **8.1** sitúa al aplicativo móvil firmemente dentro del Rango Objetivo. Esto indica que el sistema se encuentra en un nivel de madurez funcional Satisfactorio, cumpliendo con las expectativas de los usuarios finales y los requisitos de negocio establecidos por el Product Owner.

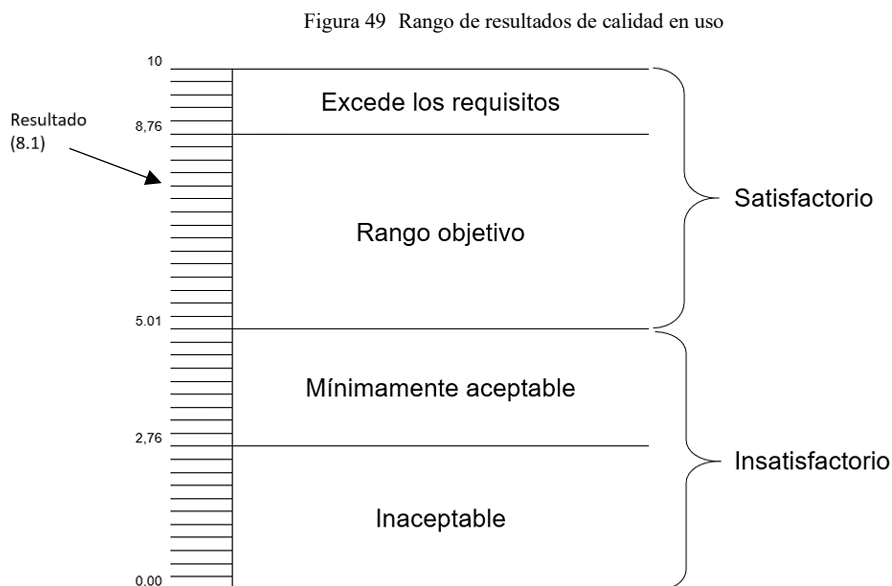


Tabla 31 Resultados Finales de Calidad en Uso - Fase 2

Modelo de Calidad en Uso						
Característica	Sub Característica	Peso de característica	Peso de Sub Característica	Medición	Resultado	
Eficacia	Tareas completas	42%	16%	0.978	15.65%	41.05%
	Objetivos logrados		16%	0.976	15.62%	
	Tareas sin errores		10%	0.978	9.78%	
Eficiencia	Tiempo de tareas	32%	16%	0.610	9.76%	19.60%
	Eficiencia del tiempo		16%	0.615	9.84%	
Satisfacción	Utilidad	26%	10%	0.832	8.32%	20.32%
	Confianza		8%	0.700	5.60%	
	Comodidad		8%	0.800	6.40%	
Total						80.98%

CONCLUSIONES

La implementación de la Fase 2 del servicio "Portafolio Estudiante" permitió dar continuidad al trabajo iniciado en la Fase 1, incorporando nuevas funcionalidades planificadas sobre la arquitectura limpia de tres capas y la infraestructura de microservicios ya establecida, lo que confirmó que las decisiones arquitectónicas tomadas en la fase anterior fueron acertadas y escalables, dado que ningún módulo existente se vio comprometido durante el proceso de integración. La gestión del desarrollo bajo el marco ágil SCRUM, organizado en seis sprints temáticos, permitió entregas incrementales y ordenadas, facilitando la validación continua de cada funcionalidad con el Product Owner y asegurando que los módulos entregados respondieran a las necesidades académicas reales de los estudiantes de la UTN. La evaluación de la calidad en uso, realizada con 70 estudiantes de la carrera de Software siguiendo la norma ISO/IEC 25022, arrojó un resultado de 80.98%, ubicando a la aplicación en el rango satisfactorio; la eficacia fue la dimensión más destacada con un 97.8% en tareas completadas y 97.6% en objetivos logrados, lo que evidencia que las funcionalidades implementadas son intuitivas y funcionalmente correctas, mientras que la eficiencia temporal obtuvo un 61% y 62%, reflejando que los usuarios nuevos aún requieren más tiempo que los expertos para completar las tareas, lo cual es esperable considerando que se trata de funcionalidades recién incorporadas; finalmente, la satisfacción alcanzó un 83.2% en utilidad percibida y un 80% en comodidad, respaldados por un coeficiente alfa de Cronbach de 0.858, confirmando la aceptación positiva del sistema por parte de la comunidad estudiantil y validando la coherencia metodológica con la Fase 1.

RECOMENDACIONES

Se recomienda mantener actualizadas de forma continua las dependencias y librerías utilizadas en el desarrollo de la aplicación móvil, especialmente las relacionadas con el framework Flutter y los paquetes asociados a cada uno de los módulos implementados a lo largo de las dos fases del proyecto, dado que el ecosistema de desarrollo móvil evoluciona constantemente y las versiones desactualizadas pueden generar incompatibilidades, vulnerabilidades de seguridad o comportamientos inesperados que comprometan el correcto funcionamiento de las funcionalidades ya entregadas, por lo que establecer un proceso periódico de revisión y actualización de dependencias contribuirá a la estabilidad y longevidad del sistema.

Se recomienda aplicar a futuro herramientas de IA dentro de la aplicación para mejorar el seguimiento académico. Se pueden aplicar sistemas que detecten de forma automática si una tarea fue hecha con inteligencia artificial o al menos detectar un porcentaje en la que fue usada. Esto entrega a los profesores datos que les permite saber sobre la originalidad de cada actividad. De esta forma, se fomentaría más la cultura de trabajo honesto y responsable entre los estudiantes.

A medida que el número de usuarios activos en la plataforma crezca y la demanda sobre los servicios aumente, se recomienda implementar una estrategia de escalado horizontal tanto a nivel del servidor como de los microservicios que conforman la API, distribuyendo la carga de trabajo entre múltiples instancias de forma controlada para evitar la saturación del sistema y garantizar la disponibilidad y el rendimiento del servicio especialmente en periodos de alta concurrencia, como fechas de evaluaciones a docentes o postulaciones de beca.

Finalmente, se recomienda realizar evaluaciones periódicas sobre la experiencia de los estudiantes con la aplicación móvil, aplicando la misma metodología de calidad en uso basada en la norma ISO/IEC 25022 utilizada en este trabajo, de manera que sea posible comparar los resultados obtenidos entre las distintas fases del proyecto, identificar áreas de mejora, detectar posibles regresiones en la experiencia del usuario y tomar decisiones fundamentadas en evidencia que orienten la evolución continua del sistema hacia estándares de calidad cada vez más altos.

REFERENCIAS

- [1] S. Fu, K. Xue, M. Yang, and X. Wang, "An exploratory study on users' resistance to mobile app updates: Using netnography and fsQCA," *Technol. Forecast. Soc. Change*, vol. 191, p. 122479, Jun. 2023, doi: 10.1016/J.TECHFORE.2023.122479.
- [2] Ministerio de Telecomunicaciones y de la Sociedad de la Información, "Ecuador ya cuenta con su Agenda de Transformación Digital 2025 – Ministerio de Telecomunicaciones y de la Sociedad de la Información." Accessed: Apr. 08, 2025. [Online]. Available: <https://www.telecomunicaciones.gob.ec/ecuador-ya-cuenta-con-su-agenda-de-transformacion-digital-2025/>
- [3] C. Bröhl, P. Rasche, J. Jablonski, S. Theis, M. Wille, and A. Mertens, "Desktop PC, tablet PC, or smartphone? an analysis of use preferences in daily activities for different technology generations of a worldwide sample," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10926 LNCS, pp. 3–20, 2018, doi: 10.1007/978-3-319-92034-4_1/TABLES/3.
- [4] D. F. Marcelo Santiago, "Implementación del servicio 'portafolio estudiante' en la plataforma móvil de la universidad técnica del norte utilizando el marco de trabajo SCRUM y la norma ISO/IEC 25022," Feb. 2025, Accessed: Mar. 28, 2025. [Online]. Available: <https://repositorio.utn.edu.ec/handle/123456789/16969>
- [5] R. A. Huertas Burgos, "Desarrollo de una guía metodológica para mejorar la documentación del capítulo técnico de los trabajos de grado de la carrera de Software que utilicen la metodología SCRUM," Mar. 2024, Accessed: Apr. 14, 2025. [Online]. Available: <https://repositorio.utn.edu.ec/handle/123456789/15812>
- [6] Trejos-Gil Carlos Andrés, Meriño Córdoba Víctor Hugo, Landinez Martinez Daniel Alfredo, and Vega-Royero Shirley, "ISO/IEC 25022:2016 - Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use." Accessed: Apr. 01, 2025. [Online]. Available: <https://www.iso.org/standard/35746.html>
- [7] Universidad Técnica del Norte, "Plan Estratégico de Desarrollo Institucional (PEDI) 2021 - 2025," 2021.
- [8] U. Nations, "Objetivos de Desarrollo Sostenible | Naciones Unidas", Accessed: Apr. 01, 2025. [Online]. Available: <https://www.un.org/es/impacto-acad%C3%A9mico/page/objetivos-de-desarrollo-sostenible>
- [9] I. A. Lescano Vásquez, "Desarrollo de una aplicación móvil utilizando el framework Flutter para fomentar el área turística del GAD de Pedro Moncayo," Nov. 2022, Accessed: Mar. 28, 2025. [Online]. Available: <https://repositorio.utn.edu.ec/handle/123456789/13228>
- [10] J. Carlos and M. Pila, "Aplicación de un Framework para la evaluación de la calidad en uso del SII-Académico de la Escuela Politécnica Nacional," Feb. 2020, Accessed: Mar. 29, 2025. [Online]. Available: <http://bibdigital.epn.edu.ec/handle/15000/21367>

- [11] E. A. Maldonado Naranjo and L. E. Mejía Gutiérrez, “Creación de aplicación Web y Mobile para administración de reservas de aulas y laboratorios para la Universidad Salesiana,” 2024, Accessed: Apr. 29, 2025. [Online]. Available: <http://dspace.ups.edu.ec/handle/123456789/27864>
- [12] N. Thallapally and N. Thallapally, “Designing scalable and robust microservice architectures for modern applications,” *International Journal of Science and Research Archive*, vol. 13, no. 2, pp. 4140–4145, Dec. 2024, doi: 10.30574/IJSRA.2024.13.2.2232.
- [13] H. Ünlü, D. E. Kennouche, G. K. Soylyu, and O. Demirörs, “Microservice-based projects in agile world: A structured interview,” *Inf. Softw. Technol.*, vol. 165, p. 107334, Jan. 2024, doi: 10.1016/J.INFSOF.2023.107334.
- [14] Honorable Consejo Universitario de la Universidad Técnica del Norte, “Reglamento de Portafolio Universitario,” 2014, Accessed: Jun. 03, 2025. [Online]. Available: https://legislacion.utn.edu.ec/wp-content/uploads/2017/12/2.REGLAMENTO_PORTAFOLIO_UNIVERSITARIO-1.pdf
- [15] I. Reascos, F. Garrido, C. Pineda, F. Salazar-Fierro, R. Pomasqui, and J. Cachipuendo, “Evaluation of the Integrated University Information System at Universidad Técnica del Norte Using the DeLone and McLean Success Model,” *Data and Metadata*, vol. 4, Jan. 2025, doi: <https://doi.org/10.56294/dm2025187>.
- [16] J. L. Escobar-Reynel *et al.*, “Modelo de desarrollo para la construcción de aplicaciones móviles educativas,” *Tecnológicas*, vol. 24, no. 52, pp. 110–135, Dec. 2021, doi: 10.22430/22565337.2065.
- [17] C. Lara and L. Figueroa, “Metodología ágil para el desarrollo de aplicaciones móviles educativas,” 2020.
- [18] Universidad Técnica del Norte, “Plan Estratégico de Desarrollo Institucional (PEDI) 2021-2025,” 2024. Accessed: Jun. 09, 2025. [Online]. Available: <https://transparencia.utn.edu.ec/wp-content/uploads/2024/01/PEDI-2021-2025.pdf>
- [19] IBM, “What Are Microservices? | IBM.” Accessed: Jun. 09, 2025. [Online]. Available: <https://www.ibm.com/think/topics/microservices>
- [20] Microsoft, “API design - Azure Architecture Center | Microsoft Learn.” Accessed: Jun. 09, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/azure/architecture/microservices/design/api-design>
- [21] Microsoft, “Resistencia y alta disponibilidad en microservicios - .NET | Microsoft Learn.” Accessed: Jun. 09, 2025. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/architecture/microservices/architect-microservice-container-applications/resilient-high-availability-microservices>
- [22] TypeScript, “TypeScript: Handbook - The TypeScript Handbook.” Accessed: Jun. 09, 2025. [Online]. Available: <https://www.typescriptlang.org/docs/handbook/intro.html>
- [23] P. Vogel, “TypeScript - Understanding TypeScript | Microsoft Learn.” Accessed: Jun. 09, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2015/january/typescript-understanding-typescript>

- [24] NestJS, "Documentation | NestJS - A progressive Node.js framework." Accessed: Jun. 09, 2025. [Online]. Available: <https://docs.nestjs.com/>
- [25] Microsoft, "API gateways - Azure Architecture Center | Microsoft Learn." Accessed: Jun. 10, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/azure/architecture/microservices/design/gateway>
- [26] AWS, "What is Amazon API Gateway? - Amazon API Gateway." Accessed: Jun. 10, 2025. [Online]. Available: <https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>
- [27] Microsoft, "Soberanía de datos por microservicio - .NET | Microsoft Learn." Accessed: Jun. 10, 2025. [Online]. Available: <https://learn.microsoft.com/es-es/dotnet/architecture/microservices/architect-microservice-container-applications/data-sovereignty-per-microservice>
- [28] Microsoft, "Designing the infrastructure persistence layer - .NET | Microsoft Learn." Accessed: Jun. 10, 2025. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/infrastructure-persistence-layer-design#the-repository-pattern>
- [29] TypeORM, "Getting Started | TypeORM." Accessed: Jun. 10, 2025. [Online]. Available: <https://typeorm.io/docs/getting-started/>
- [30] IBM, "What Is Mobile Application Development? | IBM." Accessed: Jun. 11, 2025. [Online]. Available: <https://www.ibm.com/think/topics/mobile-application-development>
- [31] AWS, "¿Qué es Flutter? - Explicación de la aplicación Flutter - AWS." Accessed: Jun. 11, 2025. [Online]. Available: <https://aws.amazon.com/what-is/flutter/>
- [32] Dart, "Dart overview | Dart." Accessed: Jun. 12, 2025. [Online]. Available: <https://dart.dev/overview>
- [33] Dart, "dart compile | Dart." Accessed: Jun. 12, 2025. [Online]. Available: <https://dart.dev/tools/dart-compile>
- [34] Flutter, "Hot reload | Flutter." Accessed: Jun. 12, 2025. [Online]. Available: <https://docs.flutter.dev/tools/hot-reload>
- [35] Flutter, "Building user interfaces with Flutter." Accessed: Jun. 13, 2025. [Online]. Available: <https://docs.flutter.dev/ui>
- [36] Flutter, "Widget catalog." Accessed: Jun. 13, 2025. [Online]. Available: <https://docs.flutter.dev/ui/widgets>
- [37] Flutter, "Performance | Flutter." Accessed: Jun. 09, 2025. [Online]. Available: <https://docs.flutter.dev/perf>
- [38] M. Robert C, "Clean Architecture," 2018, Accessed: Jul. 20, 2025. [Online]. Available: <https://agorism.dev/book/software-architecture/%28Robert%20C.%20Martin%20Series%29%20Robert%20C.%20Martin%200->

%20Clean%20Architecture_%20A%20Craftsman%E2%80%99s%20Guide%20to%20Software%20Structure%20and%20Design-Prentice%20Hall%20%282017%29.pdf

- [39] R. Martin, "Clean Coder Blog." Accessed: Jun. 09, 2025. [Online]. Available: <https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>
- [40] K. Schwaber and J. Sutherland, "La Guía de Scrum." Accessed: Jun. 10, 2025. [Online]. Available: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- [41] M. Patel, "Iterative vs Incremental Development Models - 2024." Accessed: Jun. 14, 2025. [Online]. Available: <https://www.aishco.com/iterative-vs-incremental-development-models/>
- [42] Sentionio, "Metodologías Agile: los 4 valores y 12 principios del Manifiesto Ágil." Accessed: Jun. 10, 2025. [Online]. Available: <https://sentionio.io/blog/valores-principios-agile-manifiesto-agil/>
- [43] Atlassian, "Scope creep isn't quite the enemy you think it is - Work Life by Atlassian." Accessed: Jun. 13, 2025. [Online]. Available: <https://www.atlassian.com/blog/productivity/scope-creep>
- [44] Atlassian, "Crafting an Effective Product Release Plan in 10 S... - Atlassian Community." Accessed: Jul. 20, 2025. [Online]. Available: <https://community.atlassian.com/forums/App-Central-articles/Crafting-an-Effective-Product-Release-Plan-in-10-Steps/ba-p/2868770>
- [45] OWASP Foundation, "OWASP Mobile Top 10 | OWASP Foundation." Accessed: Jun. 10, 2025. [Online]. Available: <https://owasp.org/www-project-mobile-top-10/>
- [46] OnlyFlutter, "How to Store Data in Secure Storage in Flutter - OnlyFlutter." Accessed: Jun. 13, 2025. [Online]. Available: <https://onlyflutter.com/how-to-store-data-in-secure-storage-in-flutter/>
- [47] CyberArk, "Certificate Pinning ." Accessed: Jul. 20, 2025. [Online]. Available: <https://www.cyberark.com/es/what-is/certificate-pinning/>
- [48] Owasp, "Prácticas de Codificación Segura - Guía rápida de referencia | Prácticas de Codificación Segura | OWASP Foundation." Accessed: Jul. 20, 2025. [Online]. Available: <https://owasp.org/www-project-secure-coding-practices-quick-reference-guide/stables-es/02-checklist/05-checklist>
- [49] M. Rodríguez Véliz, A. Hernández González, R. S. Lima, and Y. N. Musa, "Validación de un modelo de protección basado en la ofuscación del código," *Ingeniería Industrial*, vol. 45, no. 2, pp. 91–108, 2024, Accessed: Jul. 20, 2025. [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59362024000200091&lng=es&nrm=iso&tlng=es
- [50] M. Ogata, J. Franklin, J. Voas, V. Sritapan, and S. Quiroigico, "Withdrawn NIST Technical Series Publication Warning Notice Withdrawn Publication Superseding Publication(s) (if applicable) Title Vetting the Security of Mobile Applications Additional Information (if applicable)," 2019, doi: 10.6028/NIST.SP.800-163r1.

- [51] Dart Team, "Error handling | Dart." Accessed: Jun. 10, 2025. [Online]. Available: <https://dart.dev/language/error-handling#exceptions>
- [52] New Relic Inc., "Monitor tu aplicación móvil Flutter | New Relic Documentation." Accessed: Jun. 10, 2025. [Online]. Available: <https://docs.newrelic.com/es/docs/mobile-monitoring/new-relic-mobile-flutter/monitor-your-flutter-application/>
- [53] W. Cunningham, "Manifiesto for Agile Software Development." Accessed: Jun. 14, 2025. [Online]. Available: <https://agilemanifesto.org/>
- [54] J. Highsmith, "History: The Agile Manifesto." Accessed: Jun. 14, 2025. [Online]. Available: <https://agilemanifesto.org/history.html>
- [55] "Principles behind the Agile Manifesto," 2001. Accessed: Jun. 14, 2025. [Online]. Available: <https://agilemanifesto.org/principles.html>
- [56] A. Alliance, "What is Agile? | Agile 101." Accessed: Jul. 20, 2025. [Online]. Available: <https://agilealliance.org/agile101/>
- [57] M. Tamzid, "Agile Methodologies - Scrum." Accessed: Jun. 10, 2025. [Online]. Available: <https://viblo.asia/p/agile-methodologies-scrum-GrLZDAanlk0>
- [58] ISO 25000, "NORMAS ISO 25000." Accessed: Jul. 20, 2025. [Online]. Available: <https://www.iso25000.com/index.php/normas-iso-25000>
- [59] ISO 25000, "ISO 25010." Accessed: Jul. 20, 2025. [Online]. Available: <https://www.iso25000.com/index.php/normas-iso-25000/iso-25010>
- [60] ISO/IEC, "ISO/IEC 25000:2014(en), Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE." Accessed: Jul. 20, 2025. [Online]. Available: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25000:ed-2:v1:en>
- [61] ISO/IEC, "ISO/IEC 25022:2016 - Systems and software engineering — Systems and software quality requirements and evaluation (SQuaRE) — Measurement of quality in use." Accessed: Jul. 20, 2025. [Online]. Available: <https://www.iso.org/standard/35746.html>
- [62] ISO/IEC, "ISO/IEC 25040:2011 - Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Evaluation process." Accessed: Jul. 20, 2025. [Online]. Available: <https://www.iso.org/standard/35765.html>
- [63] F. Leandro and E. Rosero, "Implementación del servicio 'Portafolio Docentes' en la plataforma móvil de la Universidad Técnica del Norte a través del marco de trabajo Scrum y la Norma ISO/IEC 25022," May 2025, Accessed: Jun. 21, 2025. [Online]. Available: <https://repositorio.utn.edu.ec/handle/123456789/17222>
- [64] J. C. De Almeida, F. Brito E Abreu, and D. S. De Almeida, "Cross-Platform Mobile App Development: The IscteSpots experience," *Proceedings - 2023 38th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2023*, pp. 11–16, 2023, doi: 10.1109/ASEW60602.2023.00006.

- [65] C. Wang, H. Zhang, and X. Feng, "Research on the Development of a Campus Comprehensive Service App for College Students," in *2024 IEEE First International Conference on Data Intelligence and Innovative Application (DIIA)*, IEEE, Nov. 2024, pp. 1–7. doi: 10.1109/DIIA62678.2024.10871379.
- [66] K. M. Reddy, M. Sobhana, R. S. Pavan, and A. Kanagala, "ProfBook: Mobile Application for Streamlining Venue Management in Academic Institutions," *10th International Conference on Advanced Computing and Communication Systems, ICACCS 2024*, pp. 680–685, 2024, doi: 10.1109/ICACCS60874.2024.10717076.
- [67] Mike. Cohn, "Agile Estimating and Planning - Mike Cohn - Google Libros." Accessed: Dec. 09, 2025. [Online]. Available: https://books.google.com.uy/books?id=BuFWHffRJssC&printsec=frontcover&source=gsbs_atb#v=onepage&q&f=false
- [68] A. Joshi, S. Kale, S. Chandel, and D. Pal, "Likert Scale: Explored and Explained," *Br. J. Appl. Sci. Technol.*, vol. 7, no. 4, pp. 396–403, Jan. 2015, doi: 10.9734/bjast/2015/14975.
- [69] J. Brooke, "SUS - A quick and dirty usability scale," 1986, Accessed: Dec. 12, 2025. [Online]. Available: <https://hell.meiert.org/core/pdf/sus.pdf>
- [70] H. C. Oviedo and A. Campo-Arias, "Aproximación al uso del coeficiente alfa de Cronbach," *Rev. Colomb. Psiquiatr.*, vol. 34, no. 4, pp. 572–580, 2005, Accessed: Dec. 12, 2025. [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0034-74502005000400009&lng=en&nrm=iso&tlng=es

ANEXOS

Anexo A. Fotografías de aplicación del taller práctico y encuesta SUS



FACULTAD	FICA
NOMBRE	

INDICACIONES:

1	Registre en la tabla con la hora de inicio y la hora de fin de cada tarea. (Ctrl + Shift + F)
2	Indique si la aplicación mostró algún error durante la tarea (SI/NO)
3	Indique si completó la tarea con éxito (SI/NO)
Nota:	Inicie cada objetivo desde la pantalla principal (Calendario de actividades)

NOTA: Todas las tareas se realizarán en el Portafolio Estudiante (Abrir UTN Móvil → Ingresar al Portafolio Estudiante)

Objetivo	Nro.	Descripción de la tarea	Inicio	Fin	Total	Errores	Finalizó	Observación
Ver comunicados	1	Al acceder al Portafolio Estudiante gestionar Comunicados: 1. Visualizar enlace del comunicado importante. 2. Visualizar el comunicado y contestar la encuesta obligatoria. 3. Visualizar contenido del comunicado informativo.			0:00:00			
Ver pagos	2	En el menú principal (parte inferior), seleccionar la opción Más y luego Pagos: 1. Deslizar hacia abajo para visualizar la lista de todas sus transacciones. 2. Seleccionar/expandir un pago para ver su detalle. 3. Seleccionar la opción Factura o Matrícula para visualizar el documento.			0:00:00			
Ver formatos de vinculación	3	En el menú principal (parte inferior), seleccionar la opción Más y luego Formatos Vinculación: 1. Deslizar hacia abajo para visualizar la lista de todos los formatos. 2. Seleccionar opción "Ver" de cualquier formato del listado. 3. En el modal, elegir Abrir con (visualizar en una aplicación compatible) o Descargar (guardar en la carpeta de descargas).			0:00:00			
Ver contenidos de la materia	4	En el menú principal, seleccionar la opción Registro Académico y luego Aula Virtual de una materia: 1. Seleccionar en el apartado Más (la última pestaña de la barra superior) y luego Contenidos . 2. Presionar sobre los títulos de las unidades para desplegar u ocultar los temas. 3. Usar el menú de tres puntos (?) para expandir o contraer todos los contenidos de una sola vez.			0:00:00			
Comentar en Foro General	5	En el menú principal, seleccionar la opción Registro Académico y luego Aula Virtual de una materia: 1. Seleccionar en el apartado Más y luego Foro General . 2. Seleccionar (+) botón de crear (esquina inferior derecha). 3. Escribir un comentario y luego Enviar comentario.			0:00:00			
Completar una evaluación online	6	Buscar y seleccionar la actividad de tipo evaluación en el Calendario de Actividades : 1. Presionar el botón Iniciar Evaluación , leer y aceptar el aviso con las indicaciones para rendir la evaluación. 2. Responder las preguntas de opción única/múltiple, emparejamiento y respuesta abierta que estén disponibles. 3. Finalizar el intento, enviar la evaluación y verificar la nota obtenida.			0:00:00			
Evaluar el desempeño de su docente	7	Desde el menú principal, seleccionar la opción Más y luego Evaluación Docentes : 1. Escoger un docente que esté disponible para ser evaluado. 2. Llenar el formulario evaluación respondiendo todas las preguntas. 3. Presionar en Finalizar Evaluación , confirmar el envío del formulario y verificar el mensaje de confirmación exitosa.			0:00:00			
Final		Una vez finalizado el taller, guárdelo y complete la encuesta. https://forms.office.com/P/UNEZEHDK			0:00:00			

Anexo C. Tabulación del taller práctico

ID	T1 Tiempo	T2 Tiempo	T3 Tiempo	T4 Tiempo	T5 Tiempo	T6 Tiempo	T7 Tiempo	Tiempo Total	Errores	Tareas logradas	Objetivos logrados	Reclamos
1	01:40	01:21	02:11	00:42	01:20	01:51	04:23	13:28	1	18	6	NO
2	02:53	01:12	01:50	00:53	01:39	03:32	00:36	12:35	0	21	7	NO
3	03:09	00:34	03:27	00:46	02:02	04:09	01:15	15:22	0	21	7	NO
4	03:33	01:56	00:44	00:45	02:28	05:11	01:26	16:03	0	21	7	NO
5	01:38	01:01	00:06	01:18	00:19	03:23	00:39	08:24	0	21	7	NO
6	02:20	00:50	00:30	01:01	00:46	02:33	02:42	10:42	0	21	7	NO
7	02:01	00:39	02:21	02:44	01:06	02:38	01:37	13:06	0	21	7	SI
8	01:39	00:38	00:43	01:12	00:57	02:26	00:15	07:50	0	21	7	SI
9	02:20	00:50	00:30	01:01	00:46	02:33	02:42	10:42	0	21	7	NO
10	01:29	00:51	00:39	00:57	03:18	02:04	00:25	09:43	0	21	7	NO
11	03:47	03:27	02:42	01:37	01:22	03:42	01:23	18:00	2	15	5	SI
12	01:15	00:50	00:57	00:50	00:55	01:53	00:57	07:37	0	21	7	NO
13	00:31	00:33	00:23	00:34	01:28	01:29	00:35	05:33	0	21	7	SI
14	01:38	01:49	01:16	01:05	03:10	03:47	00:51	13:36	0	21	7	SI
15	04:06	01:04	01:08	01:57	02:38	01:56	00:57	13:46	0	21	7	NO
16	02:12	03:03	02:23	02:05	05:32	03:05	00:49	19:09	0	21	7	SI
17	01:28	01:06	00:22	00:38	00:32	01:02	00:41	05:49	0	21	7	NO
18	01:40	01:21	02:11	00:42	01:20	01:51	01:23	10:28	0	21	7	NO
19	03:21	05:40	01:22	01:05	03:17	01:52	01:28	18:05	1	18	5	SI
20	02:05	01:10	01:06	00:40	00:29	03:31	00:49	09:50	0	21	7	NO
21	01:57	01:12	02:56	01:01	03:43	05:28	00:59	17:16	0	21	7	NO
22	02:15	01:38	01:27	01:35	00:38	00:24	00:56	08:53	0	21	7	NO
23	03:11	00:50	00:40	00:41	00:38	01:33	01:42	09:15	0	21	7	NO

24	01:12	00:39	00:21	00:23	00:49	01:54	00:35	05:53	0	21	7	NO
25	02:49	02:41	01:50	01:03	02:48	04:09	00:49	16:09	1	18	6	SI
26	03:44	00:54	02:32	02:00	03:38	01:20	01:18	15:26	0	21	7	SI
27	01:29	01:21	00:35	00:57	01:49	01:08	01:00	08:19	0	21	7	NO
28	03:16	00:52	01:03	01:07	00:48	03:38	00:59	11:43	0	21	7	NO
29	01:00	04:51	01:06	01:50	01:33	02:32	01:29	14:21	1	18	6	SI
30	01:32	00:53	00:33	00:58	00:27	01:25	00:39	06:27	0	21	7	NO
31	01:32	01:13	01:29	02:27	01:11	03:51	01:05	12:48	0	21	7	NO
32	00:38	01:00	03:05	01:58	00:56	04:00	01:11	12:48	0	21	7	SI
33	01:46	01:15	01:18	01:53	01:19	03:44	01:17	12:32	0	21	7	NO
34	01:38	01:01	00:06	01:18	00:19	03:23	00:39	08:24	0	21	7	NO
35	03:16	00:52	01:03	01:07	00:48	03:38	01:36	12:20	0	21	7	NO
36	02:20	00:50	00:30	01:01	00:46	02:33	02:42	10:42	0	21	7	NO
37	03:43	01:13	02:30	01:26	00:46	02:30	01:12	13:20	0	21	7	NO
38	02:23	00:51	00:34	01:11	01:06	01:43	02:14	10:02	0	21	7	NO
39	02:37	00:54	00:43	04:53	00:02	00:19	01:02	10:30	0	21	7	NO
40	03:03	00:28	00:39	00:54	00:35	01:25	00:35	07:39	0	21	7	NO
41	01:06	00:38	00:23	01:07	01:39	03:46	00:44	09:23	2	15	5	SI
42	01:43	00:34	00:31	00:53	00:51	04:18	01:01	09:51	0	21	7	NO
43	01:29	01:02	00:21	00:32	00:32	01:15	00:52	06:03	0	21	7	NO
44	04:14	00:53	01:21	01:18	00:46	03:16	00:21	12:09	0	21	7	NO
45	01:28	01:36	01:54	00:55	02:38	02:26	00:22	11:19	0	21	7	NO
46	02:33	01:22	01:44	01:21	10:43	00:50	00:31	19:04	0	21	7	SI
47	03:47	01:47	02:55	02:15	01:58	03:34	00:30	16:46	0	21	7	NO
48	03:01	02:25	01:54	01:11	00:55	04:44	02:23	16:33	0	21	7	SI
49	01:32	00:28	00:45	01:51	01:04	02:08	00:53	08:41	0	21	7	NO
50	01:14	00:50	01:16	00:46	00:51	02:45	01:13	08:55	0	21	7	SI
51	01:45	00:30	00:36	01:07	01:02	02:00	00:59	07:59	0	21	7	SI

52	04:09	00:37	00:38	01:38	01:49	02:50	00:40	12:21	0	21	7	NO
53	03:16	00:52	01:03	01:07	00:48	03:38	01:59	12:43	0	21	7	NO
54	02:50	01:12	00:46	00:53	01:36	02:49	00:51	10:57	0	21	7	NO
55	03:04	01:31	00:51	00:55	03:50	04:37	02:25	17:13	0	21	7	NO
56	01:42	00:46	00:40	00:48	03:48	02:37	01:01	11:22	0	21	7	NO
57	01:08	00:17	00:53	00:47	01:41	00:51	00:36	06:13	0	21	7	NO
58	01:02	01:37	01:22	01:27	00:32	01:13	01:41	08:54	1	18	6	SI
59	05:35	03:31	02:31	00:56	03:33	03:57	01:17	21:20	0	21	7	SI
60	02:06	00:48	01:16	00:36	00:18	04:05	00:42	09:51	0	21	7	NO
61	02:38	01:34	01:57	01:08	01:32	04:29	01:53	15:11	0	21	7	NO
62	08:12	01:32	01:03	03:45	01:33	02:22	07:37	26:04	1	18	6	SI
63	02:56	01:11	01:03	00:26	01:07	02:58	01:14	10:55	0	21	7	NO
64	01:00	02:18	01:03	00:55	01:05	01:56	00:07	08:24	0	21	7	SI
65	00:50	00:47	00:59	00:41	00:36	02:32	03:15	09:40	0	21	7	NO
66	01:14	01:20	00:56	01:46	01:15	01:23	01:13	09:07	0	21	7	NO
67	01:46	01:23	01:11	01:34	01:27	03:46	01:35	12:42	0	21	7	NO
68	02:13	00:42	00:41	00:49	00:31	01:14	00:54	07:04	0	21	7	NO
69	03:10	00:45	01:37	01:04	00:50	03:23	03:05	13:54	0	21	7	NO
70	04:40	00:44	00:47	00:35	00:36	02:27	02:22	12:11	1	18	6	SI

Anexo D. Tabulación encuesta SUS

Id	Facultad	Carrera FICA	Nivel de Carrera	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
1	FICA	Software	4	5	2	5	2	5	1	5	1	4	1
2	FICA	Software	4	4	2	5	2	4	2	4	2	4	2
3	FICA	Software	4	4	3	4	2	4	3	5	3	4	3
4	FICA	Software	4	5	2	5	1	4	1	5	2	5	1
5	FICA	Software	5	4	2	5	1	5	1	4	1	3	3
6	FICA	Software	5	5	2	5	1	5	1	5	1	5	1
7	FICA	Software	5	4	2	4	1	4	3	4	2	4	1
8	FICA	Software	5	4	2	4	4	4	2	4	2	4	2
9	FICA	Software	5	5	5	5	5	5	5	5	4	4	4
10	FICA	Software	5	3	2	4	2	4	3	4	2	3	3
11	FICA	Software	5	4	2	5	1	4	2	5	2	5	1
12	FICA	Software	5	4	4	3	4	3	2	4	3	3	2
13	FICA	Software	5	3	4	3	2	4	2	3	5	3	1
14	FICA	Software	5	4	3	4	1	4	2	4	2	4	1
15	FICA	Software	5	4	1	5	1	5	3	5	3	4	2
16	FICA	Software	5	5	2	5	2	5	2	3	2	4	2
17	FICA	Software	5	4	3	5	1	4	1	5	2	5	1
18	FICA	Software	5	4	3	4	3	4	3	4	4	4	4
19	FICA	Software	6	5	2	4	1	4	1	5	1	5	1
20	FICA	Software	6	4	4	2	4	4	3	4	2	2	3
21	FICA	Software	6	5	4	4	4	4	2	4	2	5	3
22	FICA	Software	6	5	4	4	4	5	3	4	4	5	3
23	FICA	Software	6	4	2	5	1	5	1	5	1	4	2
24	FICA	Software	6	4	3	3	3	3	4	3	3	4	3
25	FICA	Software	6	4	2	3	4	2	3	3	4	4	4

26	FICA	Software	6	4	3	3	1	3	3	2	3	3	1
27	FICA	Software	6	3	1	4	1	3	3	4	1	3	1
28	FICA	Software	6	5	4	4	4	4	3	5	3	4	4
29	FICA	Software	6	5	5	5	5	5	5	5	5	5	5
30	FICA	Software	6	5	2	5	1	4	1	5	1	4	1
31	FICA	Software	6	3	2	4	3	4	2	4	2	4	1
32	FICA	Software	6	5	1	5	1	5	1	5	1	5	1
33	FICA	Software	6	3	2	4	2	4	3	3	2	3	2
34	FICA	Software	6	5	1	5	1	4	1	5	1	5	1
35	FICA	Software	6	5	4	4	3	4	1	4	2	4	2
36	FICA	Software	6	4	3	4	3	4	3	4	2	4	2
37	FICA	Software	6	5	1	5	1	5	1	5	1	5	1
38	FICA	Software	6	5	2	5	1	4	1	5	1	5	1
39	FICA	Software	6	5	1	5	1	5	1	3	2	5	1
40	FICA	Software	6	2	3	4	2	4	2	4	3	5	2
41	FICA	Software	7	3	3	3	3	3	3	3	3	3	3
42	FICA	Software	7	4	3	5	2	4	1	4	2	5	1
43	FICA	Software	7	4	3	4	2	4	2	5	3	4	1
44	FICA	Software	7	5	2	5	1	4	2	5	2	4	4
45	FICA	Software	7	5	1	5	1	5	1	5	1	5	1
46	FICA	Software	7	5	1	5	1	5	1	4	1	5	1
47	FICA	Software	7	5	2	4	5	5	2	5	1	5	3
48	FICA	Software	7	4	3	4	2	4	3	4	4	4	2
49	FICA	Software	7	4	3	4	2	4	2	4	2	4	2
50	FICA	Software	7	3	3	3	3	4	2	3	2	3	2
51	FICA	Software	7	4	2	4	1	4	2	5	3	5	2
52	FICA	Software	7	5	1	5	4	4	1	5	1	5	1
53	FICA	Software	7	4	4	5	5	4	5	5	5	5	5

54	FICA	Software	8	5	2	4	1	4	1	4	1	5	1
55	FICA	Software	8	5	1	5	1	5	1	5	1	5	1
56	FICA	Software	8	4	2	5	1	4	1	5	1	5	1
57	FICA	Software	8	4	2	4	2	4	1	4	1	4	1
58	FICA	Software	8	4	1	5	1	5	1	4	1	4	1
59	FICA	Software	8	5	1	4	1	4	1	4	1	5	1
60	FICA	Software	8	4	3	4	3	4	3	4	3	4	3
61	FICA	Software	8	5	1	5	1	5	1	5	1	5	1
62	FICA	Software	8	4	5	4	4	3	4	5	4	4	5
63	FICA	Software	8	4	1	5	1	5	1	4	1	4	1
64	FICA	Software	8	4	4	4	4	4	2	4	3	4	3
65	FICA	Software	8	4	2	5	2	4	2	5	1	5	1
66	FICA	Software	8	5	2	5	2	4	1	4	2	5	1
67	FICA	Software	8	5	5	5	1	5	1	5	5	5	1
68	FICA	Software	8	4	4	4	4	4	4	4	4	4	4
69	FICA	Software	8	4	4	5	1	5	2	5	3	4	1
70	FICA	Software	8	4	2	4	1	3	2	5	2	4	2